

Руководство администратора

ASPLinux
Руководство администратора

Оглавление

Глава 1

Введение

Под администрированием в Linux понимается обычно две связанные между собой, но в целом различные задачи. Первая — это общая настройка системы, включая начальную загрузку, параметры текстовой консоли и X Window System, а также управление общесистемными ресурсами, такими, как дисковые разделы, файловые системы, учетные записи пользователей.

Вторая задача, часто называемая собственно системным администрированием, — управление сетью и серверами разного рода — настройка сетевых протоколов, обеспечение печати и отправки почты, файловых серверов и серверов приложений, web- и ftp-серверов и т.д. Вопросы поддержания целостности системы и безопасности ее также входят в компетенцию администратора.

Первая задача стоит перед пользователем любого компьютера, если на нем установлена любая UNIX-подобная ОС, вне зависимости от того, подключен он к сети или нет. Linux — система многопользовательская, и даже на индивидуальном десктопе имеется минимум два пользователя, в том числе и администратор (суперпользователь или `root`). Вторая задача — более специальная и затрагивающая в основном профессиональных сетевых администраторов. Поэтому в настоящем руководстве основное внимание будет уделено общесистемным настройкам и управлению ресурсами в масштабе индивидуального компьютера. Для глубокого изучения вопросов сетевого администрирования следует обратиться к другим источникам информации (обзор которых приведен в заключении).

Однако Linux по своей природе — сетевая ОС. И даже на локальной машине, не имеющей подключения к какой-либо сети, использует для своих внутренних нужд сетевые службы и протоколы. Поэтому в настоящем руководстве будут рассмотрены и вопросы администрирования сети, а также безопасности системы.

Дистрибутив **ASPLinux** располагает комплексом утилит для интерактивной настройки системы. Однако следует отдавать себе отчет, что работа любых утилит настройки суть не более чем замаскированное редактирование со-

ответствующих им конфигурационных файлов. Поэтому в этом руководстве основное внимание будет уделено более тонким способам конфигурирования, нежели тем, что достигаются интерактивными методами.

Конфигурационные файлы, именуемые также стартовыми файлами, или файлами ресурсов, — это, как правило, обычные текстовые файлы, доступные для правки в любом текстовом редакторе. Они делятся на общесистемные, расположенные в каталоге `/etc` и его подкаталогах, и пользовательские, размещающиеся в домашних каталогах.

Предмет настоящего руководства — общесистемные настройки и методы управления системой в целом, поскольку индивидуальные настройки пользователя описываются в «*Руководстве по установке и настройке*», а также, частично, в «*Руководстве пользователя*». Тут будут рассмотрены следующие вопросы:

- начальный загрузчик и его настройка,
- управление дисковыми разделами,
- файловая система Linux и управление файлами, а также права доступа к файлам,
- учетные записи пользователей и управление ими,
- настройка консольного режима,
- настройка X Window System,
- администрирование сети,
- вопросы безопасности.

В заключении приведен обзор дополнительных источников информации по всем затронутым в руководстве проблемам.

1.1 Информация для читателей

В случае, если вы заметили опечатку в этой книге или у вас есть предложения по улучшению ее содержания, пожалуйста, оправьте электронное письмо по адресу support@asplinux.ru с пометкой «Документация» или оставьте свой комментарий в системе отслеживания ошибок по адресу http://bugzilla.asplinux.ru/.

Глава 2

Начальный загрузчик и его настройка

В **ASPLinux** штатно предусмотрено использование одного из нескольких начальных загрузчиков — стандартных для всех Linux-систем LiLo и GRUB, а также оригинального загрузчика ASPLoader. На этапе установки один из загрузчиков устанавливается по умолчанию. Настройка LiLo документирована в экранном руководстве и была многократно описана в книгах по Linux и в сетевых ресурсах. Поэтому в настоящем руководстве будет говориться только о настройке ASPLoader и GRUB. Разумеется, не запрещается и использование различных внешних загрузчиков (например, Acronis OS Selector), однако за информацией по этим вопросам следует обратиться к соответствующим руководствам и сопровождающей электронной документации.

2.1 Настройка и установка ASPLoader

Некоторые настройки ASPLoader можно выполнить из меню во время начальной загрузки системы (рис. 2.1).

Здесь доступны:

- смена системы, загружаемой по умолчанию (меню ≪*Configuration*≫- ≪*Set as defaults&boot*≫, рис. 2.2);
- определение параметров командной строки загружаемого ядра (меню ≪*Options*≫, рис. 2.3);
- безопасное выключение питания или перезагрузка (меню ≪*Configuration*≫- ≪*Turn off power*≫).

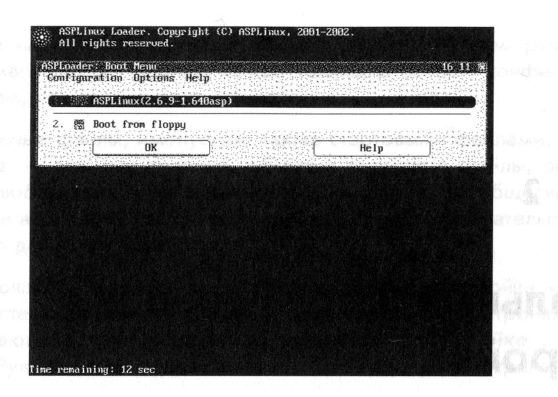


Рис. 2.1: Главное меню загрузчика ASPLoader

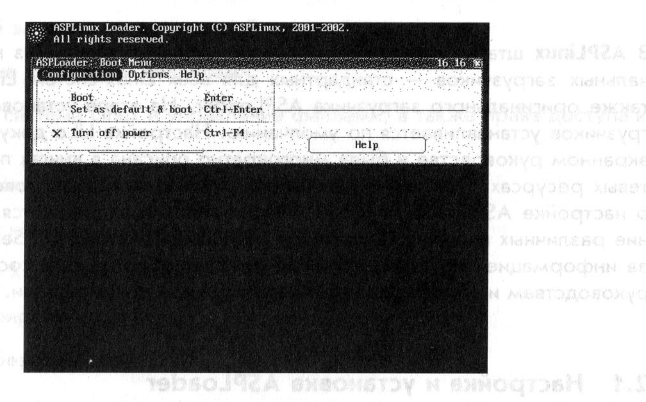


Рис. 2.2: Меню Configuration

Остальные настройки требуют редактирования основного конфигурационного файла — /etc/aspldr.conf. По умолчанию он имеет примерно следующий вид:

```
[asplinux1@ASPLinux]
icon linux
kernel /boot/vmlinuz-2.6.9-1.640asp root=/dev/sda2 ro rhgb
initrd /boot/initrd-2.6.9-1.640asp.img

[SEPARATOR]

[floppy@Boot from floppy]
icon floppy
sysboot a:
```

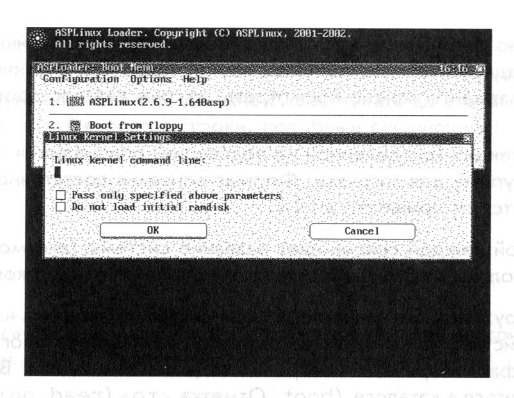


Рис. 2.3: Панель для ввода параметров ядра

```
[BOOTMGR]
video graphics
default asplinux1
timeout 15
clock 24

[ACTIVATOR]
writembr on
writeboot off
biosnum 1
mbrdev /dev/sda
language en
```

Две последние секции («BOOTMGR» и «ACTIVATOR») содержат глобальные настройки загрузчика, такие как:

- видеорежим загрузчика: кроме графического («graphics») режима по умолчанию, его можно запустить в текстовом («text») или псевдографическом («pseudographics») режимах;
- операционная система, загружаемая по умолчанию;
- время ожидания выбора загружаемой системы («timeout», в секундах);
- формат времени — 12- или 24-часовой;
- условия записи в MBR или загрузочный сектор раздела: ASPLoader может запускаться не только с первого физического диска, как LiLo, но и с любого другого;

- язык меню загрузчика; кроме английского (по умолчанию), можно выбрать поддерживаемые **ASPLinux** языки, такие как русский («ru»); доступные варианты можно посмотреть, открыв каталог /boot/aspldr.

В первой же секции (разделенной на подсекции) описываются операционные системы, доступные для загрузки. Формат описания достаточно прозрачный, хотя и отличается от принятого в LiLo.

Первой строкой каждой секции идет название системы (его можно редактировать произвольным образом) и имя выводимой при этом иконки.

Далее для загружаемых с винчестера Linux-систем приводится имя ядра и путь к нему, устройство, на котором расположен корневой каталог (то есть /, а не /root) и файл образа, с которого создается RAM-диск. Все эти файлы должны находиться в каталоге /boot. Отметка «ro» (read only) указывает, что этот раздел будет монтироваться только для чтения.

Для варианта загрузки с флоппи-диска вместо имени ядра указывается только загружаемое устройство — «sysboot a:». То же относится и к записям, отвечающим за загрузку отличных от Linux систем. Так, если **ASPLinux** устанавливался на диск с уже инсталлированной Windows 9x, будет автоматически создана секция вида:

```
[SEPARATOR]
[win@Windows 98]
icon windows
sysboot 1-1
```

Здесь в качестве параметров «sysboot» указываются номер диска и раздела на нем: обратите внимание, что нумерация и тех, и других начинается с единицы, как в приведенном примере (а не с нуля, как в LiLo).
Если в системе имеются, например, два физических диска, на втором (или, напротив, на первом) из которых установлен иной дистрибутив Linux или одна из *BSD систем, для этой системы также будет создана отдельная секция с меткой, подобной: «OS from disk99», а строка загрузки примет вид

```
sysboot #-#
```

Как уже говорилось, ASPLoader функционирует, будучи установленным на диск и раздел, отличный от первого раздела первого физического диска. Однако при автоматическом конфигурировании он в этом случае способен загрузить только **ASPLinux** и Windows, но для иных Linux-систем существует обходной путь.

Первый шаг в этом направлении — загрузить **ASPLinux** и, вслед за этим, смонтировать устройство, на котором находится каталог с ядром другой Linux-системы (скорее всего, это будут /boot или /), например:

```
mount /dev/hdb1 /mnt/linux2
```

(точка монтирования, конечно, должна существовать до этого). Затем в файл /etc/aspldr.conf вносятся строки, подобные следующим:

```
[SEPARATOR]

[Linux2]
icon linux
kernel /mnt/linux2/boot/vmlinuz_26 root=/dev/hdb1 ro
```

где указывается путь до ядра системы соответственно с ее точкой монтирования. Вслед за этим ASPLoader активизируется командой

```
/sbin/aspldr
```

что, как и для LiLo, обязательно делать после любого изменения его конфигурационного файла. Если что-либо было сделано неправильно (например, допущена ошибка в определении пути до ядра, или раздел с ним не был предварительно смонтирован), появится сообщение об ошибке. Если все в порядке, никакого видимого эффекта не последует (что, впрочем, не гарантирует, что вторую систему можно будет загрузить — детали см. ниже).

Теперь вторую файловую систему можно размонтировать и перезагрузить компьютер. При этом в меню ASPLoader появится новый пункт, соответствующий второму варианту Linux, который при соответствующем выборе и будет загружен.

2.2 Установка и настройка GRUB

В этой части руководства пойдет разговор о популярном загрузчике GRUB[1], который также поставляется с дистрибутивом **ASPLinux**.

Основные отличия GRUB помимо изначально присущих уникальных возможностей, ориентированных, в основном, на разработчиков и, как следствие, мало понятных рядовым пользователям, такие:

- принимает практически все форматы исполняемых файлов;
- обеспечивает загрузку ядер, совместимых и ограниченно совместимых со спецификацией Multiboot;
- поддерживает «цепочный» механизм для ОС и загрузчиков, которые не совместимы со спецификацией Multiboot;

[1]Сокр. от англ. «GRand Unified Bootloader».

- поддерживает загружаемые модули;
- поддерживает редактируемый текстовый конфигурационный файл;
- поддерживает различные файловые системы, такие как: FAT16 и FAT32, Ext2fs и Ext3, Reiserfs, XFS и другие;
- обеспечивает автоматическую декомпрессию сжатых gzip-файлов;
- не зависит от геометрии дисков и таким образом переход к диску с другой трансляцией номеров блоков не потребует изменения конфигурации;
- автоматически определяет LBA-режим — если BIOS поддерживает LBA, GRUB пользуется этой поддержкой;
- поддерживает сетевую загрузку по TFTP-протоколу;
- поддерживает терминальный доступ по последовательному интерфейсу, т.е. может использоваться для управления в станциях, с отсутствующей локальной консолью.

Основное отличие GRUB от других загрузчиков заключается в трех вариантах интерфейса[2]: меню и редактора секций меню (составляющих меню-ориентированную часть), а также командном, которые смогут удовлетворить любые запросы пользователей за счет различной функциональности. О них и пойдет речь в последующих главах.

2.2.1 Командный интерфейс GRUB

Командный интерфейс GRUB очень похож на `bash`, в нем присутствует память команд и автоматическое дополнение ввода. Запуск интерактивной оболочки производится из консоли путем вызова одноименной команды **grub** или клавишей `c` в момент загрузки. Так выглядит стандартное приглашение GRUB:

```
GNU GRUB  version 0.93  (640K lower / 3072K upper memory)

 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename.]

grub>
```

Информацию по любой из команд можно получить, набрав `help <имя команды>`.

[2] Эти интерфейсы могут быть доступны путем нажатия на любую клавишу в течение нескольких секунд во время загрузки экранного меню.

В простейшем случае, для установки GRUB потребуется всего две команды: root (hdX,Y) и setup (hdZ). Первая из команд указывает, где искать каталог /boot/grub, причем имя устройства всегда заключается в круглые скобки, где X — номер диска, а Y — номер раздела (Z — также номер диска, обычно равный 0, но об этом позже). Нумерация разделов начинается с нуля, что может показаться неочевидным. Тут на помощь приходит автодополнение команд:

```
grub> root (<TAB><TAB>
```

Первое нажатие клавиши Tab выдаст список допустимых устройств (или сразу же подставит устройство, если оно в системе единственное). Следующее нажатие Tab выводит на экран список всех разделов выбранного диска. Окажется полезной и команда find, которая точно укажет номер раздела, где создан искомый каталог. Например,

```
grub> find /boot/grub/stage1
```

ищет файл stage1 в каталоге /boot/grub. С помощью этой же команды можно найти любой файл на всех разделах диска или дискете. Не стоит забывать, что путь (в примере /boot/grub) — обязательный компонент имени, поэтому для файлов в корневом каталоге раздела необходимо добавить символ «/» перед их именем.

Команда setup выполнит все необходимые для инсталляции действия. В качестве параметра ей передается диск, с которого и будет происходить загрузка.

Перечень доступных команд достаточно обширен и всегда может быть выведен на экран нажатием Tab (без указания первого символа команды будет выведен весь список). Кроме команд, использование которых предполагает наличие специальных знаний (blocklist, debug, displayapm, displaymem, impsprobe, ioprobe, read, serial, setkey, terminal, testload, uppermem), имеются следующие группы команд:

- управления — см. таблицу 2.1
- работы с файлами — см. таблицу 2.2
- управления доступом — см. таблицу 2.3
- модификации разделов — см. таблицу 2.4
- настройки внешнего вида — см. таблицу 2.5

Конечно, приведенный список команд далеко не полон, более подробное обсуждение было бы слишком объемным при том, что еще не рассмотрены команды, при помощи которых и выполняются варианты загрузки. Эти же команды являются основным содержанием конфигурационного файла, о котором будет рассказано ниже.

Команда	Действие
boot	передать управление ядру, загруженному командой kernel, или «чужому» загрузчику, загруженному командой chainloader
halt	выключить компьютер
help [команда]	выдать подсказку на заданную команду
quit	выйти из GRUB
reboot	выполнить перезагрузку
pause	ожидать нажатия клавиши

Таблица 2.1: Команды управления

Команда	Действие
cat	вывести содержимое файла на экран
cmp	сравнить содержимое двух файлов

Таблица 2.2: Команды работы с файлами

Команда	Действие
password	обычно помещается в конфигурационном файле и при достижении этой команды требует ввода пароля
lock	блокировать выполнение команд для неидентифицированного пользователя

Таблица 2.3: Команды управления доступом

Команда	Действие
partnew	создать первичный раздел
partype	изменить тип раздела

Таблица 2.4: Команды модификации разделов

Команда	Действие
color	задать цвета меню
vbeprobe	определить и вывести доступные режимы видеоадаптера
testvbe РЕЖИМ	тестировать РЕЖИМ видеоадаптера

Таблица 2.5: Команды настройки внешнего вида

2.2.2 Структура конфигурационного файла

Конфигурационный файл GRUB называется `grub.conf` и располагается, как правило, в `/boot/grub`. В начале файла обычно размещаются команды задания цветов:

```
color light-gray/blue black/light-gray
```

Здесь вторая пара цветов определяет основной и фоновый цвета для выбранных позиций меню, а первая — для остальных.

Время (в секундах) от момента вывода меню до выполнения секции, определенной по умолчанию, задается командой:

```
timeout 30
```

Секция по умолчанию задается, как:

```
default 0
```

Если загрузка по умолчанию по какой-либо причине невозможна, то будет предпринята попытка выполнить секцию, указанную в команде:

```
fallback 1
```

Разумеется, цифры, определяющие секцию меню, могут быть любыми[3].

Описание каждой из секций меню начинается с команды:

```
title ТЕКСТ
```

где ТЕКСТ — остаток строки, начиная с первого непустого символа после `title`.

Обязательной командой в любой из секций меню является уже упомянутая команда root. Операционные системы, которые хотя бы частично соответствуют Multiboot Specification, загружаются командой `kernel`, причем в строке можно указывать дополнительные параметры. Таким образом команда

```
kernel (hd0,4)/boot/vmlinuz-2.6.9-1.667asp root=/dev/hda5 vga=791
```

загрузит ядро ОС Linux с раздела /dev/hda5 и подставит его же в качестве корневого для дальнейшей загрузки, а также переведет видеоадаптер в режим 1024x768 графической консоли[4].

[3]Однако нужно учитывать, что нумерация начинается с нуля.

[4]Т.н. «frame buffer mode».

Для ОС, которые не поддерживают Multiboot Specification, в первую очередь устанавливается бит активности раздела, выбранного командой `root: makeactive`, а затем по цепочке загружается собственный загрузчик указанной ОС: `chainloader +1`.

Для загрузки систем семейства Windows 9x, которые не могут быть загружены из соседних разделов (вне зависимости от флага активности грузится все равно первый из разделов), нужно использовать команды `hide` и `unhide`. Так, если первый и второй первичные разделы содержат Windows 9x, то для загрузки второй системы нужно включить в `grub.conf` следующие команды:

```
hide (hd0,0)
unhide (hd0,1)
root (hd0,1)
makeactive
chainloader +1
```

Приведенные здесь аргументы `hide`, `unhide` и `root` для загрузки такой конфигурации должны быть очевидны.

Еще одна проблема, которая может возникнуть при использовании ОС типа Windows — неспособность загружаться со второго и последующих дисков. Для ее решения применяют технику подмены[5] (от англ. «swapping technique»):

```
map (hd0) (hd1)
map (hd1) (hd0)
```

И напоследок важная рекомендация, обычно содержащаяся в инструкциях ко всем менеджерам загрузки: до установки нового менеджера загрузки следует сохранить MBR. В Linux можно воспользоваться такой командой (запись в файл `mbr-backup` на дискете):

```
dd if=/dev/hda of=/media/floppy/mbr-backup bs=512 count=1
```

2.2.3 Использование утилиты grubby

В целях конфигурирования загрузчика[6] (установка и удаление секций в файле) можно применять утилиту **grubby**.

Ниже описаны оба основных применения этой программы. Для того, чтобы создать новую секцию в конфигурационном файле `grub.conf` (он используется по умолчанию) необходимо выполнить следующую командную строку:

[5] На самом деле также поступает и BIOS.

[6] Следует обратить внимание, что описываемые здесь манипуляции также подходят и для LiLo, и для ASPLoader.

Опция	Описание параметра
–add-kernel=ЯДРО	добавить новую секцию для ЯДРА
–args=АРГУМЕНТЫ	добавить АРГУМЕНТЫ командной строки ядра, которые будут переданы ему при загрузке. Эти аргументы объединяются с шаблонными при –copy-default. Если аргумент уже присутствовал, он будет заменен новым значением.
–aspldr	использовать стиль конфигурационного файла ASPLoader
–bootloader-probe	попытаться определить используемый загрузчик
–copy-default	копировать параметры (такие как аргументы ядра и корневой раздел) из секции ядра по умолчанию
–grub	использовать стиль конфигурационного файла GRUB
–initrd=ОБРАЗ	использовать ОБРАЗ в качестве начального RAM-диска
–lilo	использовать стиль конфигурационного файла LiLo
–make-default	установить новую секцию как секцию по умолчанию
–remove-kernel=ЯДРО	удалить все секции с совпадающим ЯДРОМ
–title=ТЕКСТ	использовать ТЕКСТ как уникальный заголовок секции

Таблица 2.6: Основные опции командной строки **grubby**

```
/sbin/grubby \
        --add-kernel=/boot/vmlinuz-2.6.9-1.667asp \
        --initrd=/boot/initrd-2.6.9-1.667asp \
        --copy-default \
        --make-default \
        --title "Linux-2.6.9" \
        --args="root=/dev/hda5"
```

Для удаления секции применяется такая команда:

```
/sbin/grubby \
        --remove-kernel=/boot/vmlinuz-2.6.9-1.667asp
```

Теперь несколько слов об основных опциях командной строки для **grubby** (см. таблицу 2.6).

Об остальных опциях более детально можно узнать из руководства, поставляемого в виде одноменной man-страницы.

2.2.4 Меню-ориентированный интерфейс

Режим меню является интерфейсом по умолчанию GRUB, который отображается на этапе загрузки (рис. 2.4).

В этом режиме операционные системы или сконфигурированные секции для ядер Linux отображаются в виде списка, отсортированного по именам. Пере-

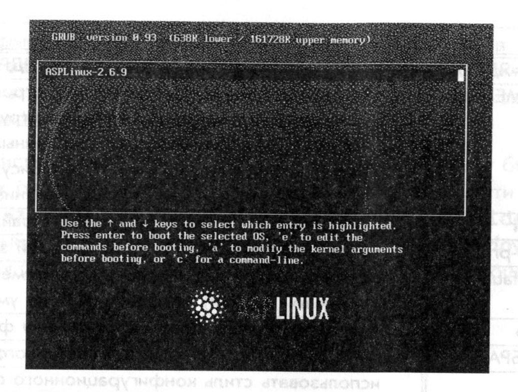


Рис. 2.4: Меню загрузчика GRUB

GRUB version 0.93 (638K lower / 161728K upper memory)

kernel /boot/vmlinuz-2.6.9-1.640asp root=/dev/sda2 rhgb
initrd /boot/initrd-2.6.9-1.640asp.img
boot
root (hd0,1)
setup (hd0)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.

ASPLINUX

Рис. 2.5: Редактор секций меню

мещение по списку осуществляется клавишами курсора (`Up` и `Down`), таким образом выбирается пункт, отличный от установленного по умолчанию. Запуск системы происходит по нажатию `Enter`. Как вариант, при установленном временном интервале ожидания по его окончании автоматически загружается система или ядро, определенное по умолчанию командой `default`.

Путем нажатия клавиши `e` будет осуществлен вход в режим редактора секций меню, а клавиши `c` — вход в командный режим соответственно.

Редактор секций меню, вызываемый по нажатию клавиши `e` и меню загрузчика позволяет модифицировать команды, описанные в выбранной секции. После входа на экране отображается содержимое секции в том виде, в котором оно было задано в конфигурационном файле, как видно на рис. 2.5.

Соответственно пользователь может изменять содержимое этих строк или добавлять новые (o добавит строки после текущей, а O — перед текущей). Удаление строки осуществляется нажатием d, в то время как редактирование — нажатием клавиши e.

После ввода всех изменений клавишей b запускается исполнение этой последовательности команд и загружается операционная система. Клавиша же Esc отменяет все изменения и перезагружает GRUB в стандартный интерфейс меню. Как уже упоминалось выше, клавиша c позволяет выбрать интерфейс командной строки.

2.3 Восстановление загрузчика

Хотя сервера находятся как правило под управлением единственной ОС, все же помимо переустановок других систем, которые, к сожалению, могут испортить данные загрузочной записи, встречаются случаи сбоя загрузчика после обновления «железа», в частности жесткого диска. В связи с этим возникает необходимость восстановления данных или установки загрузчика. Ниже описана последовательность действий, необходимая для восстановления информации загрузчика.

Вначале нужно загрузить систему с первого установочного диска в вариант «*Recovery Console*». При этом если используются SATA-диски, необходимо подгрузить модуль контроллера командой `modprobe <имя модуля>`, например:

```
modprobe sata_via
```

После этих операций стоит определить корневой раздел. Это достигается выполнением команды `fdisk -l`, которая отображает список доступных разделов жесткого диска с указанием их типа файловой системы. Корневой раздел будет одним из тех разделов, которые имеют тип, обозначенный как Linux.

Следующим шагом выполняется подгрузка драйвера этой файловой системы и его монтирование. Например, драйвер Ext3 загружается командой `modprobe ext3`, а монтирование — командой `mount -t ext3 <раздел> /mnt`. Здесь под разделом понимается имя корневого раздела в том виде, в каком его показывает `fdisk`, например, /dev/hda5. Подразумевается, что на корневом разделе используется файловая система Ext3 (по умолчанию при установке).

Предпоследний шаг заключается в выполнении следующих команд (смена корневого каталога системы, монтирование специальных файловых систем proc и sys, запуск менеджера udev):

```
chroot /mnt
```

```
mount -t proc none /proc
mount -t sysfs none /sys
/sbin/start_udev
```

И наконец, в зависимости от используемого загрузчика, выполняется одна из нижеприведенных команд (вариант установки в MBR):

```
/sbin/grub-install hd0
/sbin/aspldr -m
/sbin/lilo -b /dev/hda
```

В заключении необходимо размонтировать раздел и перегрузить систему (выход в основной корневой раздел после `chroot`, размонтирование точки `/mnt`, перезагрузка системы):

```
exit
umount /mnt
reboot
```

Глава 3

Webmin

Webmin — это программа администрирования сервера, выполненная по модульной технологии. Существует базовая программа и набор модулей, при помощи которых происходит управление различными программами, установленными на Linux сервере. Модули распространяются в файлах с расширением .wbm. После установки Webmin содержит большое количество стандартных модулей.

Webmin позволяет администраторам, впервые работающим с Linux, управлять системой в стиле Windows, когда настройки всей системы расположены в одной программе. Однако, необходимо понимать, что Webmin во многих случаях не позволяет проводить «тонкую» настройку. И, в дальнейшем, после изучения особенностей Linux, рекомендуется самостоятельно изменять конфигурационные файлы программ.

Если при установке **ASPLinux** был установлен Webmin, он автоматически запускается после старта системы. Для работы с Webmin применяют любой WEB-браузер, поддерживающий таблицы и формы. Если будет использоваться модуль «*Менеджер файлов*» — браузер должен поддерживать Java апплеты.

Для подключения к Webmin в браузере следует ввести особый адрес[1].

Первая страница, которая будет выведена в браузере — это страница аутентификации, которая представлена на рис. 3.1. В соответствующих полях введите пользователя root и его пароль. После этого нажмите на кнопку «**Login**».

[1] Обычно это http://localhost:10000 для локального подключения или http://<server>:10000 для удаленного.

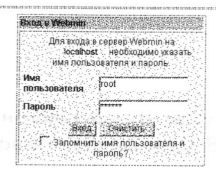


Рис. 3.1: Страница аутентификации Webmin

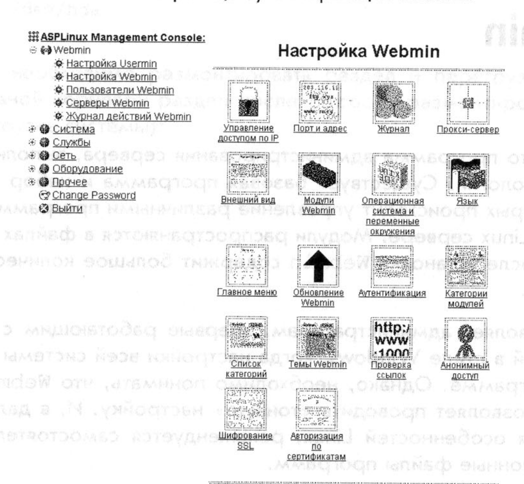


Рис. 3.2: Настройка Webmin

3.1 Настройка Webmin

Первое, что необходимо сделать после установки Webmin — настроить его (рис. 3.2). Для этих целей используйте раздел «*Настройка* Webmin».

Webmin поддерживает русскоязычный интерфейс. Выбор языка производится в разделе «*Change Language and Theme*». В списке «*Personal Choice*» выберите «*Russian KOI8 (RU_SU)*». Дальнейшее описание Webmin в данном руководстве предполагает, что был выбран русский язык интерфейса.

После установки Webmin позволяет обращаться к нему с любых компьютеров в сети. Это потенциальная уязвимость в системе безопасности. Желательно указать, с каких компьютеров можно осуществлять подключение к Webmin. В разделе «*Управление доступом по IP*» выберите «*Разрешить доступ только с перечисленных адресов*». В списке укажите IP адреса или имена компью-

теров (по одному в строке) с которых можно подключаться к Webmin. Если управление будет происходить только с компьютера, на котором установлен Webmin, необходимо указать IP адрес loopback интерфейса: 127.0.0.1.

По соображением безопасности желательно точно указать IP адрес интерфейса, который будет «слушать» Webmin. В разделе «*Порт и адрес*», в пункте «*Прослушиваемые IP адреса*» введите этот IP адрес около строки «*Only address*». Если предполагается, что управление будет происходить с различных машин, необходимо выбрать пункт «*Все*». Также можно изменить номер порта, который открывает Webmin. Для изменения порта в поле «*Прослушиваемый порт*» введите новый номер.

Еще один раздел настройки Webmin, который влияет на безопасность — это «*Аутентификация*». В нем можно настроить дополнительные параметры аутентификации. В том числе, при помощи пункта «*Блокировать доступ с компьютеров после <5> неверных попыток входа на <60 секунд>*» можно осложнить подбор паролей пользователей. Также полезно включать пункт «*Автоматически отключать пользователя после < > минут бездействия*». Если пользователь по каким-либо причинам не вышел из Webmin и оставил окно браузера открытым, то он будет автоматически отключен от Webmin. Попытка выполнения новых действий в окне браузера приведет к появлению экрана входа в систему. Пункт «*Всегда запрашивать имя пользователя и пароль*» должен быть включен всегда, даже если работа с Webmin осуществляется только с сервера, на котором Webmin установлен.

Если доступ к Webmin осуществляется по сети, желательно шифровать сеанс связи, так как пароли, передаваемые по http протоколу, не шифруются и злоумышленники могут их получить. Для шифрования сеанса связи Webmin позволяет использовать библиотеку OpenSSL. Также в системе обязательно должен быть установлен модуль Net::SSLeay языка Perl (обеспечивается установкой дистрибутивного rpm-пакета perl-Net-SSLeay), при помощи которого Webmin работает с SSL. Настройка SSL осуществляется в пункте «*Шифрование SSL*». Для работы SSL необходимо наличие файлов с ключами. В стандартной поставке Webmin поставляется файл ключа, который можно использовать для работы. Если включена поддержка SSL, URL доступа к Webmin должен начинаться с «https». Браузер, который будет использоваться при доступе к Webmin, также должен поддерживать SSL.

Во время работы Webmin может вести протокол действий, который, как правило, заносится в отдельные файлы журнала:

```
/var/webmin/miniserv.log
/var/webmin/webmin.log
```

В разделе «*Журнал*», в окне настройки можно: включить/отключить ведение журнала, заносить в журнал действия конкретного пользователя или всех пользователей, заносить действия, произведенные со всеми модулями или только с выбранными модулями и т.д.

Журналы имеют одно неприятное свойство — они постоянно растут. Если возникнет необходимость в их очистке, в настройке Webmin необходимо включить пункт «*Очищать журнал каждые XXX часов*». Для просмотра журналов Webmin следует воспользоваться модулем «*Журнал действий* Webmin», который находится в разделе «Webmin». В этом модуле можно выбрать различные критерии поиска в файлах журналов.

Некоторым модулям требуется доступ в интернет для загрузки различных файлов. И если компьютер, на котором запущен Webmin, находится за сетевым экраном (**firewall**), возможно, потребуется указать прокси–сервер, используемый в сети. Для этих целей служит раздел «*Прокси-сервер*».

После установки нового программного обеспечения на сервере, для того, чтобы этими программами можно было управлять при помощи Webmin, может возникнуть необходимость в добавлении новых модулей. Работа с модулями реализована в разделе «*Модули* Webmin». Установка модулей производится различными способами, в том числе и с http и ftp серверов.

Для работы модулей Webmin необходимо точно указать, какая операционная система используется на сервере, а также пути, в которых будет производиться поиск программ (переменная $PATH) и библиотек ($LD_LIBRARY_PATH). Возможно, потребуется ввести новые переменные среды окружения. Для управления перечисленными параметрами служит пункт «*Операционная система и переменные окружения*».

Со временем будут появляться новые версии Webmin. Для обновления версии используется раздел «*Обновление* Webmin». Также можно воспользоваться встроенной в **ASPLinux** утилитой обновления **yum**. Если будет использоваться стандартный модуль обновления, свежую версию Webmin необходимо брать не с официального сайта http://www.webmin.com. Для этих целей лучше воспользоваться ftp://ftp.asplinux.ru/pub/i386/updates/, поскольку на сервере **ASPLinux** находится адаптированная версия Webmin.

Для удобства работы с Webmin модули разделены на категории. На экране будут показаны все модули, установленные в Webmin. Напротив каждого модуля присутствует список, в котором можно выбрать категорию, к которой будет принадлежать соответствующий модуль.

Для добавления новых категорий или переименования уже существующих служит пункт «*Список категорий*».

3.2 Пользователи Webmin

Одно из полезных свойств Webmin — возможность создания новых пользователей Webmin (рис. 3.3), которым можно делегировать управление отдельными модулями.

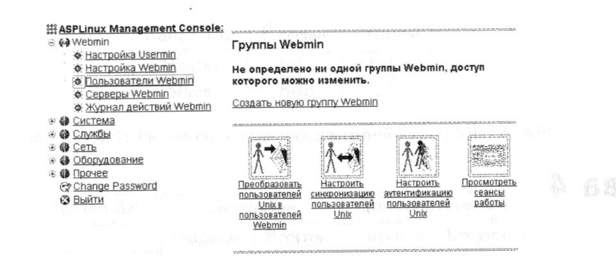


Рис. 3.3: Пользователи Webmin

При добавлении нового пользователя можно указать IP-адреса машин, с которых этот пользователь будет работать с Webmin, а также выбрать модули, которые он может использовать. Пользователь, который создается в Webmin, не добавляется к пользователям Linux. Эта особенность очень удобна, так как нет необходимости обучать пользователей Webmin работе с Linux, а также этим пользователям не разрешено входить на сервер при помощи других программ (**ssh**, **telnet**).

Существует другой способ добавления пользователей Webmin — конвертация существующих пользователей Linux в пользователей Webmin. Но для того, чтобы это стало возможным, в первую очередь необходимо создать группу пользователей Webmin. При создании группы также можно указать, с какими модулями могут работать пользователи, входящие в группу. При конвертации нет необходимости добавлять всех существующих в Linux пользователей, достаточно выбрать только тех, кто действительно будет работать с Webmin.

Глава 4

Управление дисковыми разделами и сменными носителями

Управление дисковыми разделами тесно связано с настройкой начального загрузчика, поскольку каждая из загружаемых ОС находится, как правило, в своем разделе. Так как диски, их первичные и логические разделы, а также любые другие носители суть файлы, входящие в иерархию каталогов (и локализованные в каталоге /dev), для них существует единая система номенклатуры.

4.1 Номенклатура накопителей и их разделов

Приводы гибких дисков в Linux именуются /dev/fd0 (дисковод A:) и /dev/fd1 (дисковод B:). Именование сменных носителей типа т.н. супердискет (LS-120 и их аналоги), подключаемых к интерфейсу IDE, описывается правилами для IDE-накопителей.

Любые носители информации с интерфейсом IDE/EIDE/ATAPI обозначаются в форме hdL#, где L — литера, однозначно идентифицирующая носитель физически, а # — номер раздела (первичного или расширенного) на нем. Так, первый жесткий диск на первом канале IDE (Primary Master) обозначается как /dev/hda, второй диск (или любой иной накопитель) на первом канале (Primary Slave) — как /dev/hdb, первый накопитель на втором канале (Secondary Master) — как /dev/hdc, второй накопитель на втором канале (Secondary Slave) — как /dev/hdd (Таблица 4.1). Буквенный идентификатор IDE-накопителей любого типа (жестких дисков, приводов CD ROM/R/RW, Zip, LS-120) возрастает в зависимости не от количества подключенных устройств, а от его нахождения в структуре IDE-каналов. Так, единственный винчестер, подключенный ко второму каналу в качестве первого (Secondary Master) устройства, будет именоваться /dev/hdc, даже если ранее никаких накопителей подключено не было.

Канал IDE	Первый	Второй
Master	hda	hdc
Slave	hdb	hdd

Таблица 4.1: Номенклатура накопителей IDE/EIDE/ATAPI

Канал IDE	Основной		Дополнительный	
	Первый	Второй	Первый	Второй
Master	hda	hdc	hde	hdg
Slave	hdb	hdd	hdf	hdh

Таблица 4.2: Номенклатура накопителей IDE/EIDE/ATAPI при наличии дополнительного контроллера

Накопители, подключенные к дополнительному контроллеру IDE или IDE-RAID (типа HighPoint или Promise, вне зависимости, встроенному в материнскую плату или реализованному в виде платы расширения), будут иметь более высокий буквенный идентификатор, чем любые накопители на внутреннем контроллере, даже если в BIOS установлен их приоритет по отношению к контроллеру чипсета.

Так, в конфигурации с CD-ROM и Zip на первом канале внутреннего IDE-контроллера, свободным вторым каналом и жестким диском как первым устройством на первом канале дополнительного контроллера последний будет именоваться /dev/hde. Исключение — если второй встроенный контроллер IDE отключен на уровне установок BIOS, в этом случае этот диск получит идентификатор /dev/hdc (Таблица 4.2). Дисковые разделы IDE-устройств обозначаются цифрами после буквенного идентификатора. При этом за первичными разделами (Primary Partition, которых на физическом диске не может быть более четырех) зарезервированы цифры от 1 до 4. Например, если первый диск на первом IDE-канале разбит на четыре первичных раздела, они будут обозначаться как /dev/hda1, /dev/hda2, /dev/hda3, /dev/hda4 (Таблица 4.3).

Логические тома (Volume) внутри расширенного раздела (так называемый Extended Partition) получают номера, начиная с пятого, вне зависимости

Раздел	Primary Master	Primary Slave	Secondary Master	Secondary Slave
Первый	hda1	hdb1	hdc1	hdd1
Второй	hda2	hdb2	hdc2	hdd2
Третий	hda3	hdb3	hdc3	hdd3
Четвертый	hda4	hdb4	hdc4	hdd4

Таблица 4.3: Номенклатура первичных разделов на накопителях IDE

от количества первичных разделов (и даже если ни одного первичного раздела на диске не имеется). При этом сам расширенный раздел, выступающий в виде контейнера для вложенных в него томов, получает один из номеров, закрепленных за первичными разделами.

Например, при разбиении диска на три логических раздела (тома) они будут именоваться /dev/hda5, /dev/hda6, /dev/hda7, а содержащий их расширенный раздел — /dev/hda1 (для случая с первым диском на первом канале IDE). Если же диск разбит на один первичный раздел FAT32 и один расширенный раздел Linux с четырьмя логическими разделами, номенклатура разделов будет выглядеть следующим образом: /dev/hda1, /dev/hda2, /dev/hda5, /dev/hda6, /dev/hda7, /dev/hda8. Следует подчеркнуть, что номенклатура эта ни в коей мере не зависит от файловых систем, созданных на разделах: она охватывает дисковые разделы для любых ОС (MS DOS, Windows, Linux, FreeBSD, OpenBSDи т.д.).

Информацию о разделах жестких дисков можно получить с помощью команды fdisk с указанием физического устройства в качестве аргумента, например:

```
fdisk /dev/hde
```

Далее после появления приглашения

```
Command (m for help):
```

следует дать команду p (от print), ответом на которую будет вывод информации о диске и его разделах:

```
Disk /dev/hde: 255 heads, 63 sectors, 2482 cylinders
Units = cylinders of 16065 * 512 bytes
Device     Boot  Start   End  Blocks    Id  System
/dev/hde1   *    1       3    24066     83  Linux
/dev/hde2        4       35   257040    82  Linux swap
/dev/hde3        36      2481 19647495  83  Linux
```

Впрочем, тот же результат может быть получен командой:

```
fdisk -l /dev/hde
```

или

```
fdisk -ls /dev/hde
```

Очевидно, что сменные носители типа CD-ROM будут идентифицироваться только последовательностями символов без цифр: /dev/hdc, /dev/hdd и

т.д., так как разделов на них обычно не бывает. Новые, из коробки, диски ZIP имеют раздел вида /dev/hdc4, отформатированный в vfat.

Для накопителей с интерфейсом SCSI система номенклатуры несколько иная. Жесткие диски SCSI именуются (в порядке подключения к шине) /dev/sda, /dev/sdb и так далее, их разделы — /dev/sda1, /dev/sda2, /dev/sda5 и т.д. Правила для нумерации первичных и логических разделов — те же, что и для IDE-дисков.

Номенклатура, принятая для накопителей SCSI, распространяется и на IDE-устройства при включенной эмуляции через этот интерфейс протокола SCSI, например, на записывающие и перезаписывающие CD устройства (без такой эмуляции запись на устройства ATAPI CD-R/RW невозможна) или накопители Zip.

Так, накопитель Zip в режиме эмуляции SCSI будет обозначен как /dev/sda4 (при сохранении фабричного форматирования). Накопитель же CD-ROM (а также CD-R/RW) получит обозначение /dev/scd0.

Диски с интерфейсом SATA и внешние накопители с интерфейсом USB (flash-диски, внешние жесткие диски, флоппи-диски) имеют такие же имена устройств, как и диски scsi - /dev/sdX.

4.2 Создание разделов и файловых систем

Разделы на жестком диске, куда устанавливается **ASPLinux**, создаются в ходе инсталляции системы (как это было описано в руководстве по установке). При подключении к системе нового диска на нем также следует создать разделы.

Делается это упомянутой выше командой fdisk. Например, если в компьютер с единственным винчестером на первом IDE-канале был установлен второй диск (на второй канал в качестве Master-устройства), для создания разделов на нем следует набрать в командной строке

```
fdisk /dev/hdc
```

и, по выводе приглашения этой программы, перейти к дальнейшим действиям, список которых можно получить командой m. Главные из них — следующие:

```
a toggle a bootable flag - сделать раздел загрузочным
d delete a partition - удаление раздела
l list known partition types - список поддерживаемых файловых систем
m print this menu - вывод настоящей справки
n add a new partition - создание нового раздела
p print the partition table - вывод существующей таблицы разделов
q quit without saving changes - выход из программы без сохранения изменений
```

```
t change a partition's system id - изменение идентификатора файловой системы
u change display/entry units - изменение единиц измерения объема разделов
   (с цилиндров размером 16065*512 байт на секторы размером 512 байт
v verify the partition table - проверка таблицы разделов
w write table to disk and exit - запись изменений и выход из программы
x extra functionality (experts only) - дополнительные функции
```

Из полученной справки можно видеть, что раздел на диске создается командой n (от new). Вслед за ее вводом будет последовательно предложено определить тип раздела — e (extended) или p (primary), его номер, начальный цилиндр раздела, затем конечный его цилиндр; вместо последнего можно задать размер раздела в Мбайт или Кбайт (в форме +9999M или +9999K, соответственно).

Далее для раздела, при необходимости, следует определить файловую систему, задаваемую ее шестнадцатиричным номером. Узнать номер для нужной системы можно по списку, выводимому командой l (от list). В этом списке можно увидеть, кроме файловой системы для Linux, 83) и его раздела подкачки (Linux Swap, 82), файловые системы почти всех существующих ОС (FAT16, FAT32, BSD, QNX и т.д.). Однако, не рекомендуется создавать для них разделы средствами fdisk для Linux — они не всегда будут опознаны соответствующими ОС. Кроме того, разделы для чужих ОС должны быть не только созданы, но и отформатированы.

Так что фактически, средствами fdisk лучше создать только разделы Linux и разделы подкачки.

Закончив разбиение диска, следует сохранить изменения и выйти из программы fdisk командой w — до ее подачи разделы можно перекраивать как угодно, и всегда есть возможность командой q выйти, не сохраняя созданных разделов. В Linux в общем случае не требуется перезагрузки компьютера для того, чтобы сделанные изменения структуры разделов вступили в силу. Необходимость перезагрузки возникает только тогда, когда был внесены изменения на жесткий диск, какой-нибудь раздел которого используется (примонтирован).

После разбиения диска на вновь созданных разделах следует создать файловые системы. Это осуществляется с помощью команды mkfs.ext3. В качестве параметров указываются опции форматирования (-c — с проверкой на поврежденные блоки, -v — с выдачей сообщений), а в качестве аргумента — имя раздела. Например, команда

```
mkfs.ext3 /dev/hdc1
```

создаст файловую систему Linux на первом разделе диска, подключенного как Master ко второму каналу IDE.

Файловые системы Linux можно создавать на дисках Zip — точно так же, как на жестких дисках, и на дискетах:

```
mkfs.ext3 -c /dev/fd0
```

Не следует с помощью этих команд форматировать дискеты под MS DOS для обмена данными с Windows-компьютерами: для этого существует специальный набор инструментов mtools, предназначенный для работы с дискетами формата MS DOS.

Кроме разделов с обычными файловыми системами, используемыми для хранения программ, данных и прочего, в Linux существует также понятие раздела подкачки (swap-раздела). Он создается следующим образом:

- командой `fdisk` для него выделяется место на диске, то есть создается раздел (например, `/dev/hdc2`), которому присваивается номер 82 (Linux Swap);
- командой mkswap `/dev/hdc2` (при желании — с опцией `-c`, то есть проверкой на испорченные блоки) на этом разделе создается соответствующая файловая система;
- командой swapon `/dev/hdc2` созданный раздел подкачки активизируется.

4.3 Монтирование файловых систем

Файловые системы на вновь созданных разделах должны быть смонтированы, то есть включены в иерархию каталогов общей файловой системы, начинающейся с корневого (/) каталога. Делается это командой mount, аргументами которой являются имя раздела и точка монтирования, определяющая его положение в структуре каталогов. Так, команда

```
mount /dev/hdc1 /media/disk
```

смонтирует созданный ранее раздел Linux в точку `/media/disk`, то есть он будет выглядеть как подкаталог в каталоге `/media`. Подкаталог `/media/disk` для монтирования должен быть создан заблаговременно, например, командой

```
mkdir /media/disk
```

Возможно монтирование не только устройства с файловой системой Linux, но и многих других файловых систем. Тип файловой системы, как правило, распознается автоматически. Если это по каким-либо причинам не произошло, следует указать его в явном виде (с помощью параметра `-t`). Так, команда `mount` с параметром `-t msdos` смонтирует раздел с файловой системой

FAT16, с параметром -t vfat — раздел для Windows 9x/ME, с параметром -t ufs — раздел для FreeBSD или OpenBSD. Для монтирования диска CD-ROM требуется параметр -t iso9660.

Аналогично монтируются и сменные накопители — дискеты или Zip:

```
mount /dev/fd0 /media/floppy
```

или, соответственно,

```
mount /dev/hdd /media/zip100.0/
```

Как и в случае с дисковыми разделами, при необходимости следует указать тип файловой системы, например, для, спасательной resque-дискеты Linux, команда монтирования приобретет вид

```
mount -t ext3 /dev/fd0 /media/floppy/
```

а для Zip-диска с фабричной разметкой

```
mount -t vfat /dev/hdd /media/zip100.0/
```

При совместном использовании **ASPLinux** и Windows с помощью команды mount можно получить доступ к разделам с файловой системой FAT16 или FAT32 (обратная процедура тоже возможна, но требует сторонних Windows-программ explore2fs или Itools, еще недостаточно надежных). Однако если на разделе Windows имеются имена файлов, содержащие символы кириллицы, для их корректного воспроизведения в Linux команду монтирования следует дать в виде

```
mount -o iocharset=utf8,codepage=866 /dev/hda1 /media/windows
```

где значение опции -o codepage=866 — это кодировка файловой системы MS DOS для имен с символами кириллицы, а iocharset=utf8 — это кодировка представления имен файлов в Linux.

Эпизодически используемые дисковые разделы и сменные носители по истечении надобности в них следует размонтировать командой umount точка_монтирования (именно так, без буквы n в названии команды, не обязательно с указанием имени устройства). Причем для сменных носителей это — обязательная процедура перед извлечением их из привода.

Впрочем, неразмонтированный диск CD-ROM извлечь и не удастся до выполнения команды

```
umount /media/cdrom
```

Однако дискеты можно удалить из привода без размонтирования. В результате целостность файловой системы может быть нарушена, результатом чего будет порча данных на ней. И потому дискеты следует не только размонтировать командами

```
umount /media/floppy
```

но желательно и убедиться, что процесс этот был успешно завершен. Проще всего это делается командой

```
mount (без параметров)
```

Если дискета была успешно размонтирована, каталог /media/floppy будет отсутствовать в списке смонтированных ФС.

Внимательного отношения требуют также Zip-приводы. Если такие устройства с LPT- или SCSI-интерфейсом, подобно CD-ROM, извлечь из привода без размонтирования не удается, то для Zip-дисков с интерфейсом IDE такое вполне возможно: выброс диска для них блокируется (после команды mount) только при включении режима эмуляции SCSI через IDE.

Все смонтированные устройства (в том числе и сменные) автоматически размонтируются при корректном останове системы (известной комбинацией клавиш Ctrl+Alt+Del, командами reboot, halt или shutdown) — никакой порчей данных это не грозит. Диски аудио-CD не содержат файловую систему и в монтировании, и размонтировании не нуждаются.

4.4 Настройка постоянно используемых файловых систем

Разделы, на которых располагается сам **ASPLinux** и регулярно используемые данные, должны быть доступны постоянно. Поэтому они монтируются автоматически в ходе загрузки системы. Список таких устройств и условия их монтирования описываются в файле /etc/fstab[1]. Содержимое его имеет примерно следующий вид:

```
# <file system> <mount point>   <type>   <options>          <dump> <pass>
/dev/hda1       /               ext3     defaults             0      1
/dev/hda2       none            swap     swap                 0      0
/dev/cdrom      /media/cdrom    auto     owner,noauto,ro      0      0
/dev/fd0        /media/floppy   auto     owner,noauto         0      0
```

[1] fstab - от англ. File System TABle - таблица файловых систем.

```
/dev/hdd         /media/zip100.0 auto      noauto,owner     0     0
proc             /proc     proc      defaults          0     0
none             /dev/pts          devpts    gid=5,mode=620   0     0
```

За исключением двух последних строк, описывающих виртуальные файловые системы, служащие для взаимодействия с ядром Linux, и второй строки, активизирующей раздел подкачки, остальные отвечают за монтирование реальных устройств. Первое поле каждой записи — имя файла соответствующего устройства (или, в некоторых случаях, его псевдоним), второе — точка его монтирования.

Для сменных устройств это обуславливает возможность монтирования их вводом сокращенной команды, для CD-ROM, например, имеющей вид

```
mount /dev/cdrom
```

без указания истинного имени файла источника и точки монтирования.

Третья запись — это тип файловой системы. Для дискового раздела Linux тип файловой системы указан явным образом («ext3»), для сменных носителей он будет определен автоматически.

Четвертое поле записи — условия монтирования устройств. Так, значение его для раздела Linux («defaults») означает, что он монтируется автоматически, в ходе загрузки системы. И по предопределенным условиям монтирования может содержать исполнимые файлы (параметр «exec»), быть доступным как для чтения, так и для записи («rw»), содержать файлы устройств («dev»), допускать асинхронный (то есть с кэшированием в оперативной памяти) ввод/вывод («async»), и т.д. Если какую-либо из этих возможностей требуется запретить, это следует сделать в явном виде либо соответствующим параметром (например, «ro» — только для чтения, «sync» — синхронный, без кэширования, ввод/вывод), либо отрицающим параметром «no*» (например, параметр «noexec» запрещает запуск исполняемых файлов с данного носителя).

Важным параметром является «suid», также автоматически включаемый при монтировании устройства как «defaults». Он означает возможность учета прав доступа к записанным на него файлам (о правах доступа будет подробно рассказано в соответствующем разделе).

Последние два поля каждой записи определяют условия резервного копирования с данных устройств («<*dump*>») и проверки их файловой системы при загрузке («<*pass*>»). Файловые системы, для которых значения этих полей равны нулю, не резервируются, и не проверяются.

Как правило, для монтирования любых устройств требуются права администратора. Однако если устройства эти внесены в fstab, в поле

«*<options>*» для них можно задать условия монтирования их обычными пользователями. Для этого служит параметр «user». Например, строки

```
/dev/cdrom /media/cdrom auto user 0 0
/dev/fd0 /media/floppy  auto user 0 0
/dev/hdd4 /media/zip100.0 auto user 0 0
```

в файле /etc/fstab указывают, что эти носители могут быть смонтированы обычным пользователем. Установка параметра «user» автоматически влечет за собой отрицание параметра «defaults». Так что если с данного носителя требуется запускать какие-либо программы или учитывать права доступа к записываемым на него файлам, соответствующие возможности должны быть указаны в явном виде. Например, строка

```
/dev/hdd4 /media/zip100.0 auto user,exec,suid 0 0
```

разрешает запуск программ и учет прав доступа для устройства Zip.

Можно разрешить автоматическое монтирование во время загрузки устройств с файловой системой, отличной от ext3, например, VFAT или NTFS. Если на устройстве имеются файлы с именами, набранными отличными от латинских символами, следует, как и при «ручном» монтировании, указать правила для их преобразования. Для раздела Windows 9x с русскими именами файлов соответствующая строка файла /etc/fstab должна иметь вид, подобный следующему:

```
/dev/hda1 /media/win vfat defaults,iocharset=utf8,codepage=866 0 0
```

Здесь предполагается, что раздел для Windows 9x расположен на первом разделе первого IDE-диска, а точка его монтирования — /media/win.

В дистрибутиве **ASPLinux** для монтирования сменных носителей по умолчанию предназначены подкаталоги каталога /media — /media/cdrom, /media/floppy, /media/zip100.0. Подкаталоги для других временно используемых файловых систем следует создать самостоятельно.

4.5 Создание разделов при помощи Webmin

Для создания разделов используется модуль «*Администратор разделов*», находящийся в разделе «*Оборудование*». В главном окне модуля показан список жестких дисков и существующих на них разделов. Также в этом модуле можно изменять параметры работы жестких дисков, но эту возможность следует использовать с большой осторожностью.

Для создания первичного раздела служит ссылка «*Добавить первичный раздел*». На появившейся странице необходимо указать тип раздела, а также его первый и последний цилиндры. После ввода значений следует нажать на кнопку «**Создать**». Расширенный раздел создается таким же образом, но при помощи ссылки «*Добавить расширенный раздел*». После создания расширенного раздела появится возможность добавлять в нем логические разделы при помощи ссылки «*Добавить логический раздел*».

Если выбрать ссылку с номером раздела на диске, будут показаны параметры выбранного раздела. В этом же окне присутствует кнопка «**Удалить**». В том случае, если раздел используется, т.е. подключен к основной файловой системе, его нельзя удалить и кнопка «**Удалить**» не будет показана. Единственное исключение — это расширенные разделы. Присутствие кнопки «**Удалить**» в параметрах этих разделов, можно отнести к недостаткам модуля «*Менеджер разделов*». Никогда не удаляйте расширенный раздел, если системой используются созданные в нем логические разделы.

По ссылке «*Изменить параметры IDE*» открывается страница настройки IDE интерфейса.[2]

Все современные жесткие диски поддерживают различные режимы DMA, поэтому пункт «*Использовать DMA*» может быть включен. Включение DMA режима значительно увеличивает скорость обмена информацией с жестким диском. Если же режимы DMA, поддерживаемые жестким диском и материнской платой неизвестны, в списке «*Режим передачи*» следует выбрать значение «*По умолчанию*», в этом случае будет установлен режим передачи данных по умолчанию.

Еще один полезный параметр, который увеличивает скорость передачи данных — «*Поддержка 32-битного ввода/вывода*», его желательно установить в «*Включить*». Также на скорость работы жесткого диска влияет пункт «*Количество секторов для многосекторного ввода/ вывода*»: большинство современных жестких дисков поддерживают максимум 16 одновременно читаемых секторов.

Для подтверждения изменений параметров диска необходимо нажать на кнопку «**Применить к диску**». После этого будет показано, с какими параметрами была запущена программа **hdparm**, а также ее вывод.

Чтобы определить, насколько увеличилась скорость обмена с жестким диском, перед внесением изменений и после этого, можно воспользоваться кнопкой **Проверка скорости**.

[2] Внимание! Параметры, которые присутствуют на этой странице, следует изменять с большой осторожностью. Есть вероятность того, что их изменение приведет к порче оборудования.

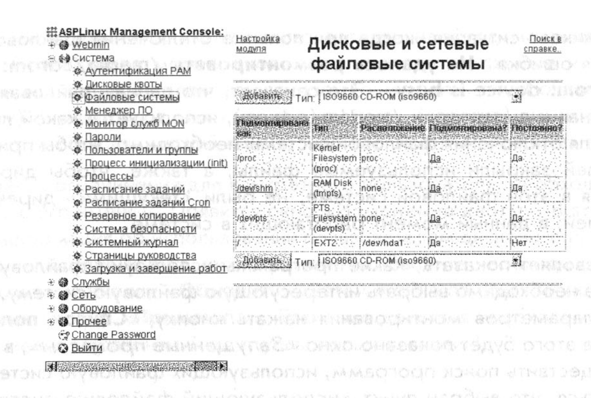


Рис. 4.1: Настройка файловых систем

4.6 Настройка файловых систем при помощи Webmin

Webmin позволяет работать с файловыми системами при помощи модуля «*Дисковые и сетевые файловые системы*», находящегося в разделе «*Система*». Модуль изменяет содержимое файла /etc/fstab, а также может монтировать и размонтировать файловые системы, описанные в этом файле.

На главной странице модуля показан список файловых систем. Для изменения параметров монтирования необходимо выбрать точку монтирования интересующей файловой системы. Также можно добавить новую файловую систему, в этом случае необходимо выбрать тип файловой системы из списка и воспользоваться кнопкой «**Добавить**».

На странице, описывающей параметры файловой системы, можно выбрать различные опции монтирования, которые совпадают с опциями, описанными в предыдущих разделах. Обратите внимание, что при возникновении необходимости монтирования файловых систем типа vfat или ntfs в которых в именах файлов используются русские буквы, опции монтирования iocharset и codepage придется добавлять в файл /etc/fstab вручную.

Если файловая система не была смонтирована при загрузке, ее можно подключить при помощи Webmin. В списке файловых систем, в столбце «*Подмонтирована?*» необходимо выбрать ссылку «*Нет*», соответствующей файловой системы. Таким же образом можно отключить файловую систему, но следует выбрать ссылку «*Да*». Другой способ подключения или отключения — воспользоваться экраном настройки файловой системы. Достаточно в пункте «*Подмонтировать сейчас?*» выбрать «*Подмонтировать*» и нажать на кнопку «**Сохранить**». Таким же образом можно отключить файловую систему, в этом случае необходимо выбрать «*Размонтировать*».

Иногда возникает ситуация, когда при попытке отключения файловой системы выдается ошибка «**Не удалось размонтировать /media/cdrom: umount: /media/cdrom: device is busy**». Это означает, что данная файловая система (подключенная к директории /media/cdrom), используется какой либо программой. Для отключения файловой системы необходимо, чтобы приложения пользователей закрыли используемые файлы, а также, чтобы директории, находящиеся в этой файловой системе, не были «текущими» директориями пользователей, в данный момент работающих в системе.

Webmin позволяет показать, какие программы используют файловую систему. В списке необходимо выбрать интересующую файловую систему, а в окне настройки параметров монтирования нажать кнопку «**Список пользователей**». После этого будет показано окно «*Запущенные процессы*», в котором можно осуществить поиск программ, использующих файловую систему. Следует убедиться, что выбран пункт «*использующий файловую систему*», а в списке выбрана необходимая файловая система. Далее надо нажать на кнопку «**Искать**». В результате поиска будет показан список процессов, использующих файловую систему. Теперь процессам, показанным в появившемся списке, можно послать сигнал завершения работы, для этого необходимо нажать на кнопку «**Завершить процесс**». В этом случае, всем процессам будет послан сигнал SIGTERM. Иногда программы игнорируют посланный им сигнал, например, если программа «зависла». В этом случае необходимо нажать на кнопку «**Снять процесс**», программам будет послан сигнал SIGKILL, по которому система прекратит их работу. После того как программы, использующие файловую систему, завершили свою работу, можно вернуться к списку файловых систем и отключить файловую систему.

4.7 Дисковые квоты

Webmin позволяет управлять дисковыми квотами. В Linux ограничение на дисковое пространство может накладываться как на отдельных пользователей, так и на группы пользователей. Ограничения могут накладываться только на физическую файловую систему, т.е. нет возможности наложить ограничения на одну директорию. Для управления дисковыми квотами используется модуль «*Дисковые квоты*», по умолчанию расположенный в разделе «*Система*».

В главном окне модуля показан список файловых систем, на которые разрешено накладывать квоты. Если список пуст, необходимо вернуться в модуль управления файловыми системами и в опциях выбранной файловой системы разрешить использование квот. После этого действия компьютер необходимо перегрузить или перемонтировать файловую систему. Для использования ограничений в файловой системе, в поле действия нажмите на ссылку «*Включить квоты*». В первом столбце таблицы будут показаны две ссылки: настройка ограничений для пользователей и для групп пользователей.

Соотвественно для ограничений пользователей необходимо выбрать ссылку «*users*». Ограничения можно вводить на количество файлов и/или блоков используемых пользователем или группой (один блок, по умолчанию равен 1 килобайту). Существуют «мягкие» и «жесткие» (строгие) лимиты. Пользователи могут превышать мягкий лимит, жесткие лимиты превысить нельзя.

Чтобы ввести ограничения для конкретного пользователя, необходимо нажать на ссылку с его именем. Появится окно «*Изменение квоты*». В этом окне показана информация об использовании пользователем дискового пространства, а также накладываемые на него ограничения. Введите необходимые параметры и нажмите на кнопку «**Обновить**». Затем в окне со списком пользователей нажмите на кнопку «**Применить**». Такие же действия необходимо выполнить при наложении ограничений на группы пользователей.[3]

[3]Внимание! Информация о квотах сохраняется в файлах `aquota.group` и `aquota.user`. Эти файлы находятся в точке монтирования файловой системы и доступны для изменения только пользователю `root`.

Глава 5

Основы управления процессами

В этом и двух следующих главах будут рассмотрены ключевые понятия, на которых базируется Linux (и все UNIX- и UNIX-подобные системы) и которые представляются наиболее непривычными пользователю, переходящему с платформы Windows — процессы, файлы и права доступа к ним, а также учетные записи пользователей. Понятия эти тесно и притом рекурсивно связаны между собой:

- пользователь запускает процесс, порождающий файл (файлы);
- права доступа для процесса определяются тем, какими правами был наделен запустивший процесс пользователь;
- файлы наделяются правами доступа и принадлежности в силу прав породившего их процесса;
- права же пользователя определяются параметрами его учетной записи.

Так что начинать рассмотрение этих понятий можно с любой точки цикла. В настоящем руководстве за точку отсчета принято понятие процесса. Следует подчеркнуть только, что по ходу описания процессов будут упоминаться (без расшифровки) понятия и атрибутов файлов, и полей учетных записей пользователя, более детально рассмотренные в следующих главах.

В качестве процесса в Linux рассматривается независимо выполняющаяся программа со своими ресурсами. Процесс может быть либо запущен пользователем (прикладные программы), либо генерироваться системой при ее работе. В последнем случае иногда говорят о т.н. виртуальных пользователях.

Каждый процесс имеет уникальный численный идентификатор (`PID, Process IDentificator`) и владельца (то есть запустившего его пользователя, реального или виртуального). Кроме того, пользовательские процессы привязаны к виртуальной консоли (терминалу), с которой они были запущены; процессы же, генерируемые системой, ни с каким терминалом не ассоциируются.

Для получения информации о протекающих процессах служит команда ps.

Запущенная неким пользователем без параметров, она выдает краткую информацию о процессах текущего терминала, владельцем которых данный пользователь является, например:

```
PID TTY      TIME CMD
309 tty1     00:00:00 bash
337 tty1     00:00:00 joe
466 tty1     00:00:00 ps
```

В приведенном примере можно видеть, что пользователем alv с первой виртуальной консоли («*TTY*»=«tty1») запущены три процесса («*CMD*») — оболочка bash, редактор joe и собственно команда ps, имеющие идентификаторы («*PID*») 309, 337 и 466.

Более полную информацию о процессах можно получить, прибегнув к различным комбинациям параметров команды ps, с которыми можно подробно ознакомиться на странице экранной документации

```
man ps
```

Так, команда ps aux позволяет получить дополнительные сведения о процессах в следующей форме:

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
Root 1 0.3 0.2 1324 524 ? S 15:50 0:07 init [3]
Root 2 0.0 0.0 0 0 ? SW 15:50 0:00 [keventd]
...
rpc 417 0.0 0.2 1468 588 ? S 15:50 0:00 portmap
...
xfs 630 0.0 1.4 4972 3676 ? S 15:50 0:00 xfs -droppriv -da ...
alv 775 0.0 0.5 2340 1304 tty6 S 15:51 0:00 -bash
alv 853 0.0 0.2 2556 736 tty3 R 16:26 0:00 ps aux
```

Наиболее существенными для дальнейшего рассмотрения являются следующие поля:

- «*USER*» — имена владельцев (включая администратора) всех процессов, запущенных в системе;
- «*PID*» — идентификатор процесса;
- «*%CPU*» и «*%MEM*» — задействованные ресурсы процессора и памяти, соответственно;
- «*TTY*» — номера терминалов, с которых запущены процессы;

- «*STAT*» — состояние процесса;
- «*NI*» — уровень приоритета процесса.

Рассмотрим подробнее эти характеристики процессов. Каждому запущенному в системе процессу, как уже говорилось, соответствует владелец, то есть запустивший его пользователь. Это не всегда реальный пользователь (или администратор) системы: в качестве владельца процесса может выступать какой-либо из стартовых сервисов. Например, в строке

```
USER PID %CPU %MEM TTY STAT START TIME COMMAND
...
xfs 287 0.0 1.6 ? S 11:44 0:00 xfs -droppriv -da
```

родительским процессом `xfs -droppriv -da` является запустивший его сервис `xfs` (сервер шрифтов X Window System).

Владелец процесса может быть обозначен его именем (`root`, `alv`, `xfs`), как в приведенных выше примерах. Однако далее можно видеть, что вместо имени может фигурировать и цифровой идентификатор пользователя (`UID`). Он совпадает с идентификатором учетной записи пользователя (о чем будет рассказано в одной из следующих глав). Для администратора `UID` всегда равен нулю, за системными демонами зарезервированы номера с 1 по 499 (в дистрибутиве **ASPLinux**). Идентификаторы реальных пользователей начинаются с номера 500 (в некоторых дистрибутивах — с 1000).

Идентификатор определяет права доступа процесса к файлам (о чем — в следующей главе). Как правило, каждый процесс наследует права доступа, которыми наделен его владелец. Однако бывают и исключения.

Забегая вперед, заметим, что помимо `UID`, процессы идентифицируются и иными способами. Это связано с тем, что иногда процессу необходим доступ к ресурсам, к которым его владелец доступа, соответственно параметрам своей учетной записи, не имеет. И тогда играют роль понятия т.н. эффективного идентификатора (`EUID`) процесса и идентификатора доступа к файловой системе (`FSUID`), используемые не только для повышения, но и для понижения полномочий процесса по сравнению с правами его владельца. Подробнее об этом будет говориться в главе об учетных записях.

Значение идентификатора процесса понятно — это его уникальный номер, при этом `PID`, равный 1, всегда имеет процесс `init`:

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
Root 1 0.0 0.2 1324 524 ? S 11:43 0:09 init
```

Это связано с тем, что процесс `init` является родоначальником для всех остальных процессов в системе, например, для процесса `mingetty`, порождающего, в свою очередь, процесс авторизации пользователей `login`:

```
F UID PID PPID PRI NI STAT TTY COMMAND
100 0 1 0 0 0 S ? init
100 0 300 1 0 0 S tty1 login -- user1
100 0 301 1 0 0 S tty2 login -- user2
100 0 302 1 0 0 S tty3 login -- user3
```

И потому каждый процесс, кроме своего собственного PID, имеет еще и идентификатор родительского процесса (PPID). В приведенном примере можно видеть, что процесс init с PID, равным 1 породил процессы с PID 300, 301 и 302 — авторизацию трех разных пользователей на трех виртуальных консолях.

Исходный же для них процесс mingetty завершился после окончания авторизации, и потому в списке не фигурирует.

Здесь же становится понятным смысл поля «*TTY*»: для процессов, привязанных к виртуальной консоли, значение его соответствует номеру последней (1, 2, 3, соответственно). Процессы же, запущенные стартовыми сервисами (демон xfs, например), ни с одним терминалом не связаны, что отражает символ ? в данном поле.

Состояние процесса отражает степень его исполнения: процесс может быть исполняемым в данное время (R), находящимся в режиме ожидания (S) или приостановленным (T), например, при помощи комбинации клавиш Ctrl+Z. В приведенном ниже примере:

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
Alv 338 0.0 0.5 2392 1372 tty2 S 12:27 0:00 -bash
Alv 670 0.0 0.4 2104 1052 tty2 T 18:10 0:00 joe admin.txt
Alv 669 0.0 0.2 2564 748 tty2 R 18:06 0:00 ps au
```

можно видеть, что процесс bash находится в ожидании (ввода команды), процесс joe приостановлен временным выходом в командную оболочку, а процесс ps исполняется в настоящее время (то есть во время отдачи этой команды).

В поле статуса может встретиться также значение Z и D. Первое соответствует т.н. процессу-«зомби» — завершившемуся дочернему процессу, от которого родительский процесс еще не принял сигнала окончания работы. По прошествии некоторого времени «зомбированные» процессы завершаются окончательно и исчезают из списка.

Символ же D означает, что процесс находится в состоянии непрерываемого ожидания (uninterruptible sleep), обычно — ввода/вывода. Такой процесс не способен завершиться сам по себе, не реагирует на системные запросы (в том числе на команды принудительного завершения) и может быть уничтожен только перезапуском системы.

Уровень приоритета процесса («NI») обозначается по-английски словом nice value (что интерпретируется обычно как степень «дружелюбия» или «тактичности» по отношению к другим процессам) и варьируется в диапазоне от -20 (минимальное «дружелюбие», то есть высший приоритет) до +20 (максимальное «дружелюбие», соответствующее низшему приоритету).

Все приведенные сведения могут потребоваться при управлении запущенными процессами. Пользователь может управлять только теми процессами, владельцем которых является. Администратор системы же располагает правами на управление всеми запущенными процессами.

Управление процессами включает в себя изменение их приоритета и принудительное завершение. Все пользовательские процессы (и большинство системных) по умолчанию запускаются с равным промежуточным приоритетом 0. И, соответственно, при многих запущенных задачах ресурсы компьютера (процессорное время и объем оперативной памяти) распределяются между ними равномерно.

При необходимости перераспределения ресурсов между программами можно изменить приоритет выполнения какой-либо из них. При этом пользователь в состоянии только уменьшить приоритет одного или нескольких процессов, владельцем которых он является, администратор же имеет возможность повысить приоритет любого процесса.

Делается это двояко. Если требуется запустить программу с приоритетом, отличным от обычного, используется команда nice с величиной изменения nice value в качестве опции (предваряемой дефисом) и именем программы в качестве опции. Так, команда

```
nice -5 joe
```

запустит редактор joe с приоритетом, уменьшенным на пять единиц. Если же опустить опцию, приоритет уменьшится на десять единиц. Для увеличения же приоритета администратор (и только он) может дать команду

```
nice --7 joe
```

что приведет к росту приоритета (то есть уменьшению «дружелюбия») на семь единиц.

Кроме того, приоритет может быть изменен для уже запущенных процессов с помощью команды renice, параметром которой является новое значение приоритета, а аргументом идентификатор (PID) процесса. Например, команда

```
renice 7 735
```

данная от лица пользователя — владельца процесса с PID 735, приведет к установке для него nice value, равного 7 (при условии, что прежний приоритет этого процесса был выше, например, 3 или 0). Администратор же может понизить приоритет того же процесса до значения nice value, равного 2, с помощью команды

```
renice 2 735
```

или присвоить ему максимальный приоритет:

```
renice -20 735
```

Необходимость ручного управления приоритетами возникает достаточно редко, а вот принудительное прерывание процесса — задача более обычная. Оно требуется, например, для выхода из безнадежно зависшей программы, не реагирующей на комбинации клавиш Ctrl+C и тому подобные. Или при невозможности закрыть окно программы в X Window System штатными средствами управления.

Для таких случаев предназначена команда kill. В общем виде в качестве опции ее используется название сигнала или его номер, а аргументом служит PID процесса. Список сигналов команды и соответствующих им номеров можно получить с помощью

```
kill -l
```

ответом на что будет

```
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR      31) SIGSYS      32) SIGRTMIN    33) SIGRTMIN+1
34) SIGRTMIN+2  35) SIGRTMIN+3  36) SIGRTMIN+4  37) SIGRTMIN+5
38) SIGRTMIN+6  39) SIGRTMIN+7  40) SIGRTMIN+8  41) SIGRTMIN+9
42) SIGRTMIN+10 43) SIGRTMIN+11 44) SIGRTMIN+12 45) SIGRTMIN+13
46) SIGRTMIN+14 47) SIGRTMIN+15 48) SIGRTMAX-15 49) SIGRTMAX-14
50) SIGRTMAX-13 51) SIGRTMAX-12 52) SIGRTMAX-11 53) SIGRTMAX-10
54) SIGRTMAX-9  55) SIGRTMAX-8  56) SIGRTMAX-7  57) SIGRTMAX-6
58) SIGRTMAX-5  59) SIGRTMAX-4  60) SIGRTMAX-3  61) SIGRTMAX-2
62) SIGRTMAX-1  63) SIGRTMAX
```

а расшифровку значений сигналов можно получить по команде

```
man 7 signal
```

Таким образом, как команда

```
kill -SIGTERM 735
```

так и команда

```
kill -9 735
```

предписывают завершить работу процесса 735. Различие их в том, что по получении сигнала SIGTERM (номер 15) программа по возможности пытается корректно завершить свою работу (с записью всех буферизованных данных), а сигнал SIGKILL (номер 9) означает немедленное и неизбежное завершение процесса.

Значение сигнала команды kill по умолчанию — -SIGTERM, и потому на практике для прекращения работы программы часто достаточно дать ее в виде

```
kill PID
```

где аргумент, как уже говорилось, определяется с помощью команды ps. Более того, кроме внешней команды /bin/kill, одноименная команда встроена и в оболочку bash (и некоторые другие, например, tcsh). И потому вместо PID процесса можно использовать номер задания оболочки

```
kill %#
```

где # определяется командой jobs (как это описано в руководстве пользователя).

Более эффективный способ отслеживания процессов и, при необходимости, управления ими — использование команды top. В отличие от команды ps, она, выведя на экран информацию о процессах (рис. 5.1), не завершает свою работу, а продолжает обновлять ее через промежуток времени, значение которого может быть установлено пользователем. Формат вывода информации также настраивается. Кроме того, возможно интерактивное управление процессами.

Доступ к справке о возможностях программы осуществляется нажатием клавиши [H] или [?] (рис. 5.2).

Основные из этих возможностей следующие:

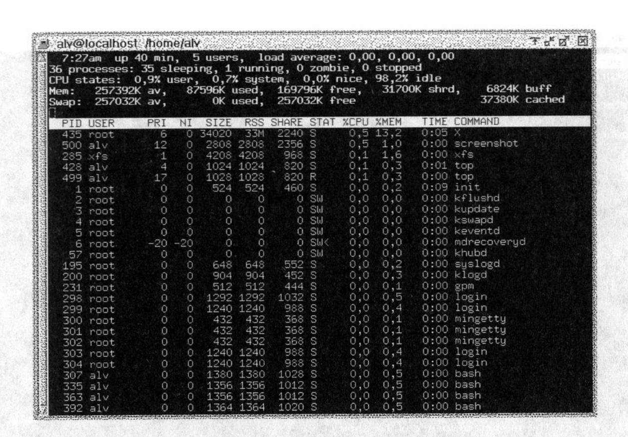


Рис. 5.1: Команда `top`, вывод на экран

- `Ctrl` + `L` — перерисовка экрана;
- `f` — удаление и добавление полей, выводимых на экран; и то, и другое осуществляется нажатием символьной клавиши-переключателя для соответствующего поля (рис. 5.3);
- `o` — изменение порядка вывода полей; для этого также используются литерные клавиши, нажатие которых в верхнем регистре приводит к смещению соответствующего поля влево, в нижнем регистре — вправо;
- `c` — показ/скрытие полных путей команд в соответствующем поле;
- `k` — снятие процесса; после нажатия этой клавиши следует предложение сначала ввести `PID` процесса, а затем — номер сигнала (по умолчанию — 15, `SIGTERM`);
- `r` — изменение приоритета процесса, для чего сначала указывается его `PID`, а затем — новое значение приоритета;
- `u` — показывать процессы только определенного пользователя, имя которого вводится после нажатия этой клавиши; если имени не ввести, будут показаны процессы всех пользователей;
- `q` — выход из программы.

Кроме того, изменяются такие параметры, как порядок сортировки (по `PID`, по возрасту, по использованию CPU и памяти), количество выводимых процессов, время (в секундах) обновления информации, и т.д. Все сделанные изменения имеют силу в текущем сеансе. Однако их можно сохранить в файле `/.toprc` нажатием клавиши `W` (верхний регистр обязателен).

Файл Сеансы Настройки Помощь

```
Proc-Top Revision 1.2
Secure mode off; cumulative mode off; noidle mode off

Interactive commands are:

space   Update display
^L      Redraw the screen
fF      add and remove fields
oO      Change order of displayed fields
h or ?  Print this list
S       Toggle cumulative mode
i       Toggle display of idle proceses
I       Toggle between Irix and Solaris views (SMP-only)
c       Toggle display of command name/line
l       Toggle display of load average
m       Toggle display of memory information
t       Toggle display of summary information
k       Kill a task (with any signal)
r       Renice a task
N       Sort by pid (Numerically)
A       Sort by age
P       Sort by CPU usage
M       Sort by resident memory usage
T       Sort by time / cumulative time
u       Show only a specific user
n or #  Set the number of process to show
s       Set the delay in seconds between updates
W       Write configuration file ~/.toprc
q       Quit

Press any key to continue
```

Новый | Терминал # 1

Рис. 5.2: Справочная система команды `top`

5.1 Управление процессами при помощи Webmin

Для управления процессами в Webmin используется модуль *«Процессы»*, который находится в разделе *«Система»* (см. рис. 5.4). Этот модуль позволяет получить список процессов, отсортированный по различным параметрам: PID, пользователь, занимаемое процессорное время или память. Также доступны различные параметры поиска процессов в системе. В результате поиска или сортировки появляется страница со списком процессов. Для того, чтобы посмотреть подробную информацию об интересующем процессе, следует нажать на ссылке с его номером.

После выбора процесса появляется страница, на которой показаны его различные параметры. Если выбранный процесс запустил на выполнение другие процессы, их список будет показан в таблице *«Дочерние процессы»*. На этой же странице предоставляется возможность изменить приоритет процесса. Достаточно в списке выбрать необходимый приоритет и нажать на кнопку «**Изменить**». Также тут можно послать сигнал процессу, для этого необходимо выбрать сигнал из списка и нажать на кнопку *«Послать сигнал»*. Чтобы корректно завершить работу процесса, ему посылают сигнал SIGTERM или нажимают на кнопку «**Завершить процесс**». Если программа зависла и/или не реагирует на посылаемые сигналы, следует нажать на кнопку «**Снять процесс**». В этом случае процессу посылается сигнал SIGKILL и операционная система уничтожает процесс. При нажатии на кнопку «**Файлы и соедине-**

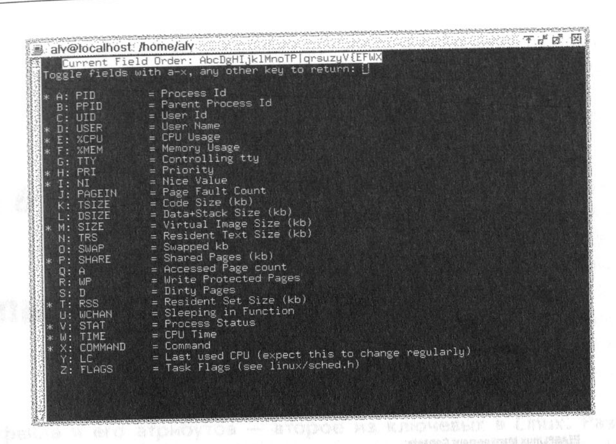


Рис. 5.3: Добавление и удаление полей вывода команды top

ния» будет сформирован список файлов, открытых текущим процессом.

Еще одна возможность, которая присутствует в модуле «*Процессы*» — запуск программ. В главном меню модуля необходимо выбрать ссылку «*Выполнить...*». Поле «*Команда для выполнения*» играет роль командной строки, в нем вводят команду с аргументами, например, `ls -l /home`. Для выполнения этой программы следует нажать на кнопку «*Выполнить*». Все, что выполняемая программа выводила бы на экран терминала, будет показано на следующей странице.

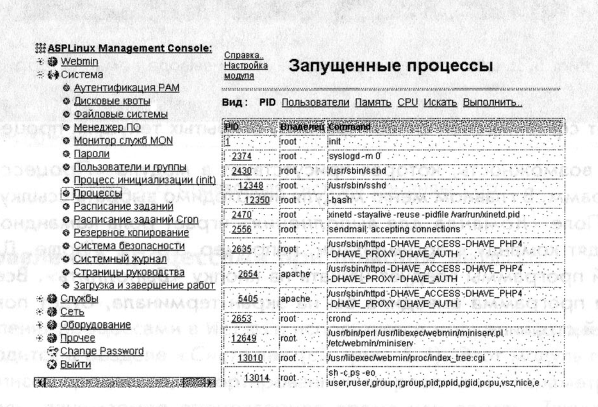


Рис. 5.4: Управление процессами при помощи Webmin

Глава 6

Файлы и их атрибуты

Понятие файла и его атрибутов — второе из ключевых в Linux. Разумеется, файлы есть и в любых других ОС, например, MS DOS или в Windows. Однако в Linux (и UNIX вообще) в качестве файлов рассматривается все, что существует в системе — исполняемые программы, созданные ими данные, дисковые и иные другие накопители, а также все прочие устройства.

Файлы в Linux организованы в виде файловых систем. Термин этот понимается в двух различных смыслах — во-первых, как физическую сущность, то есть способ хранения данных на диске (или другом накопителе), во-вторых, как логическую структуру, в которую они организованы.

Все файлы в Linux имеют набор атрибутов. В DOS/Windows таковых три — скрытый, архивный, системный. В Linux же важнейшими являются весьма многочисленные атрибуты доступа. Время создания и модификации также могут рассматриваться как атрибуты файла.

Все эти аспекты будут последовательно рассмотрены в этой главе, начиная с классификации файлов, через физическую и логическую их организацию, и заканчивая атрибутикой.

6.1 Классификация файлов

Такое разнообразие нуждается в классификации. Файлы в Linux классифицируются следующим образом:

- обычные (`regular`) файлы,
- каталоги (`directory`),
- файлы устройств (`devices`),

- специальные файлы — сокеты (sockets) и именованные каналы (named pipes),
- символические ссылки (symlinks).

Смысл понятий обычного (regular) файла понятен пользователю. Это откомпилированные бинарные программы и интерпретируемые сценарии оболочки, конфигурационные ASCII-файлы, текстовые файлы, а также файлы данных, создаваемые прикладными программами в их собственных форматах — растровой и векторной графики, текстовых процессоров и т.д. Общее между ними то, что все они могут быть непосредственно просмотрены либо стандартными командами типа cat, less, more, либо специально предназначенными для этого программами.

Каталоги (directory) — это файлы, содержанием которых является информация о входящих в их состав обычных файлах и вложенных каталогах. Это не тавтология — такой информацией, в сущности, являются просто списки имен файлов, входящих в данный каталог.

Файлы устройств соответствуют разным присутствующим в системе устройствам. Устройства эти, с одной стороны, могут быть реальными (жесткие диски, принтеры и т.д.) и т.н. псевдоустройствами, с которыми не ассоциировано никакое «железо» (например, пустое устройство /dev/null).

С другой стороны, выделяются отдельные файлы символьных устройств и блочных устройств.

К первым возможен только последовательный доступ. Примером их являются последовательные и параллельные порты. К блочным устройствам можно осуществлять произвольный доступ. Они представлены жесткими дисками и другими накопителями.

Файлы устройств идентифицируются своими номерами — основным (major) или старшим, определяющим класс устройств (например, 4 — старший номер устройств, относимых к классу терминалов, реальных и виртуальных), и дополнительным (minor), который обычно является просто порядковым номером данного устройства в своем классе.

Специальные виды файлов — каналы и сокеты — предназначены для обмена данными между процессами. Они важны для разработчиков ПО, пользователь с ними, как правило, напрямую не общается.

На символических ссылках (symlinks) следует остановиться подробнее. Они представляют собой отдаленные аналоги (и прародители) ярлыков в Windows или «теней» (shadow) в OS/2. Символические ссылки могут быть созданы командой

```
ln -s имя_файла имя_ссылки
```

на файл любого из перечисленных выше типов. Это просто именованный файл, указывающий на файл-источник, что можно определить по последнему полю вывода команды ls:

```
lrwxrwxrwx    1 root       root            4 Июн 10 15:56 /bin/sh -> bash
```

В приведенном примере это поле (/bin/sh -> bash) показывает, что файл sh представляет собой символическую ссылку на файл bash в том же каталоге /bin.

При явном обращении к символической ссылке действия (исполнение, просмотр и т.д.) осуществляются на самом деле с тем файлом, на который она ссылается.

При этом файл-источник может находиться в другом каталоге, в другом разделе диска или даже на другой машине.

Символические ссылки следует отличать от обычных, или «жестких» ссылок[1]. К последним относятся, как будет показано в следующем разделе, в том числе и имена всех файлов.

6.2 Файловая система как физическая сущность

Файловая система Linux по физической организации резко отличается от системы FAT (и VFAT, каковая — не более чем ее подмножество). Каждый файл в Linux состоит как бы из двух частей. Первая — это некая запись на диске — inode (что иногда переводится на русский как «узел»), содержащая такую информацию о файле, как его размер, формат, атрибуты (права доступа, время создания и модификации, и т.д.), но не имя. Каждый узел имеет уникальный цифровой идентификатор, по которому и отыскивается программами. Имя же файла представляет собой жесткую ссылку (hardlink, или просто link) на узел с данным идентификатором.

Отличий свойств жесткой ссылки между inode файла и его именем от символических ссылок, рассмотренных выше, несколько. Во-первых, жесткая ссылка может существовать только в том же дисковом разделе, что и inode, на который она ссылается.

Во-вторых, на один inode может быть создано произвольное количество жестких ссылок, содержание которых будет идентично между собой. То есть файл с одним и тем же физическим содержанием может выступать под целым рядом имен, и все они будут равноправны между собой: любое из них можно удалить, что не окажет никакого влияния на остальные имена файла (и, тем более, на его реальное содержимое).

[1]От англ. hardlinks

Число символических ссылок на имя файла тоже не ограничено. Однако исходное имя файла-источника и имена символических ссылок не равноправны. Конечно, имя символической ссылки может быть удалено без вреда, однако удаление имени файла-источника приведет к тому, что и все символические ссылки на него потеряют работоспособность.

Попробуем продемонстрировать все сказанное на примере. Интересующую нас информацию о файлах некоего каталога можно получить командой

```
ls -ilF -G
```

где опция -i предписывает выводить идентификаторы узлов (inode), опция -F — отличать имена каталогов от имен файлов конечным символом /, а опция -G исключает (для компактности) принадлежность файла группе, что не является пока предметом рассмотрения. Результатом команды будет нечто вроде

```
2327203 -rw-rw-r--1 62212 Июл 16 13:38 admin.txt
2327202 -rw-rw-r--1 79326 Июн 24 16:51 install.txt
2327201 -rw-rw-r--1 70578 Июл 4 09:41 qstart.txt
623501 drwxr-xr-x 3 4096 Июл 16 13:11 ris_admin/
2048748 drwxrwxr-x 5 4096 Июн 24 09:48 ris_install/
1852185 drwxr-xr-x 11 4096 Июл 13 17:19 ris_user/
279314 -rw-rw-r--1 276559 Июл1 4 19:31 user.txt
```

Первое поле каждой записи — идентификатор inode (в десятичном исчислении), второе — атрибуты файла, из которых нас интересует пока только первая позиция: символ — (дефис) означает обычный файл, символ d — каталог, символ l (который встретится позже) — символическую ссылку. Далее следует поле с количеством ссылок, связанных с данным файлом, размер (в байтах), время модификации и имя файла.

Из этого можно видеть, что в текущем каталоге присутствуют четыре файла и три подкаталога, каждый с уникальным идентификатором, различным объемом и временем модификации. Каждый из обычных файлов имеет по одной ссылке — это жесткая ссылка между его именем и inode. Число ссылок для каталогов — переменное: оно определяется количеством входящих в него подкаталогов, куда они сами входят в качестве составных элементов.

Далее, создаем жесткую ссылку на один из файлов:

```
ln admin.txt admin1.txt
```

и повторяем команду ls с теми же параметрами:

```
2327203 -rw-rw-r--2 62212   Июл 16 13:38 admin.txt
```

```
2327203 -rw-rw-r--2 62212   Июл 16 13:38 admin1.txt
2327202 -rw-rw-r--1 79326   Июн 24 16:51 install.txt
2327201 -rw-rw-r--1 70578   Июл  4 09:41 qstart.txt
623501 drwxr-xr-x 3 4096    Июл 16 13:11 ris_admin/
2048748 drwxrwxr-x 5 4096    Июн 24 09:48 ris_install/
1852185 drwxr-xr-x 11 4096 Июл 13 17:19 ris_user/
279314 -rw-rw-r--1 276559  Июл 14 19:31 user.txt
```

В результате видим, что в каталоге прибавился один файл — admin1.txt, все атрибуты которого, за исключением имени (идентификатор inode, права доступа, размер, время модификации), идентичны исходному. Одновременно для обоих файлов изменилось и количество ссылок (до двух), поскольку теперь на один узел ссылается уже два имени файла.

А теперь создадим символическую ссылку на тот же файл:

```
ln -s admin.txt admin2.txt
```

и снова выполним команду ls:

```
2327203 -rw-rw-r--2 62212 Июл 16 13:38 admin.txt
278947 lrwxrwxrwx 1 9 Июл 16 13:42 admin2.txt -> admin.txt
2327203 -rw-rw-r--2 62212 Июл 16 13:38 admin1.txt
```

Информация о файлах admin.txt и admin1.txt (как, разумеется, и о прочих файлах и каталогах) не изменилась. Однако появившийся теперь файл admin2.txt имеет другой идентификатор, размер и время доступа. Более наглядно различия между жесткими и символическими ссылками можно наблюдать при просмотре свойств файлов в Midnight Commander (рис. 6.1). Следует учесть только, что здесь идентификатор узла (пункт «*Положение*» в левой панели) дан в шестнадцатеричном исчислении.

Следует подчеркнуть, что связанные жесткой ссылкой имена файлов (в данном примере — admin.txt и admin1.txt) не являются копиями одно другого, каковые можно было бы получить, например, командой

```
cp admin.txt admin3.txt
```

выполнив которую, мы увидим с помощью команды

```
ls -ilF -G admin*
```

еще один файл того же содержания и размера

```
2327203 -rw-rw-r--2 65709 Июл 16 17:44 admin.txt
2327203 -rw-rw-r--2 65709 Июл 16 17:44 admin1.txt
278948 -rw-rw-r--1 65709 Июл 16 17:47 admin3.txt
```

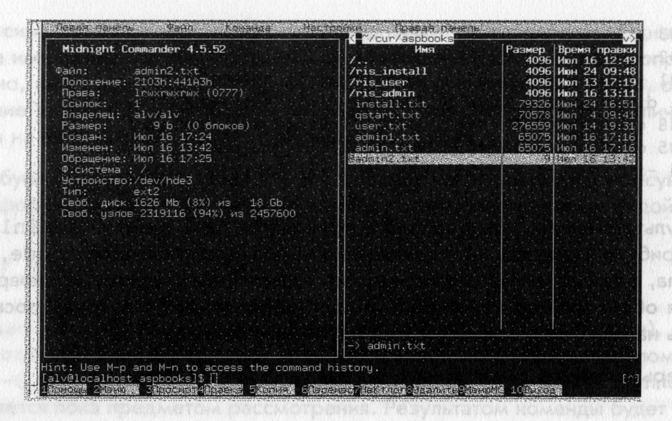


Рис. 6.1: Сравнение свойств жестких и символических ссылок

но с иным идентификатором узла и временем модификации. Идентичность связанных жесткой ссылкой файлов (и их отличие от копии любого из них) подчеркивается тем, что первые два изменяются параллельно, как в данном примере, в процессе набора настоящего текста. В чем легко убедиться, повторив команду

```
ls -ilF -G admin*
2327203 -rw-rw-r--2 66577 Июл 16 17:52 admin.txt
2327203 -rw-rw-r--2 66577 Июл 16 17:52 admin1.txt
278948 -rw-rw-r--1 65709 Июл 16 17:47 admin3.txt
```

Из ее результата можно видеть, что размер обоих файлов admin.txt и admin1.txt увеличился (за счет набора нескольких последних абзацев), изменилось и время их модификации, тогда как файл admin3.txt остался в первозданном (на момент его создания командой cp) виде.

Первым следствием такого устройства файловой системы является то, что удаление файлов в Linux происходит совершенно иначе, чем в DOS/Windows. А именно, файл считается удаленным, когда уничтожены все имена, ссылающиеся на данный inode, и закрыта последняя программа, к нему обращающаяся.

Разумеется, сами по себе данные, составляющие содержание файла, физически могут продолжать существовать на диске, но для системы они уже недоступны. А поскольку содержание файла оторвано от его имени, восстановление файла по фрагменту имени оказывается невозможным.

Пока любой файл открыт, то есть существует ссылающийся на него процесс, он продолжает существовать, даже если имя его на диске стерто, и может быть записан, скопирован, переименован, и т.д.

Второе следствие особенностей файловой системы Linux — оторванность содержания файла от его имени накладывает на это имя весьма мало ограничений.

Абсолютно запрещенными к использованию в именах файлов символами являются только / и \. Правда, некоторые другие специальные символы, такие как !, @ и прочие из верхнего ряда клавиатуры, за исключением _, всякого рода скобок и кавычек, также не рекомендуются к использованию в именах файлов, особенно в начальной позиции, но это, обычно, требование оболочки командной строки, а не системы.

Максимальная длина имени файла (включая и любое количество «расширений») — 255 знаков. А максимально возможная длина полного пути к файлу — 4096.

В Linux в общем случае файлу данных любого типа может быть приписано любое расширение (или его может не быть вовсе): на понимание его породившей программой это никак не отразится. Более того, файл может иметь несколько расширений (то есть групп знаков, разделенных точками): типичный пример — архивный компрессированный файл `*.tar.gz`.

Некоторые программы (скажем, графические редакторы или офисные пакеты) все же требуют, чтобы файл формата TIFF имел расширение `*.tif`, и т. д. Но это вызывается тем, что имя файла неявно передается программе, то есть запускающей ее команде, в качестве одного из аргументов.

6.3 Логическая организация файловой системы

Логическая организация файловой системы, то есть структура каталогов, в Linux, напротив, жестко фиксирована. Конечно, обладая правами суперпользователя, ее можно изменить. Но делать это крайне не рекомендуется — в результате система может просто утратить работоспособность.

Структура каталогов может существенно отличаться от дистрибутива к дистрибутиву. Более того, это — один из основных критериев различия главных их линий (таких, как клоны RedHat, Debian, Slackware). И потому ниже речь пойдет только о структуре каталогов дистрибутива **ASPLinux**, в значительной мере унаследованной от его прототипа — RedHat. Что, собственно, и дает основание считать его RedHat-совместимым.

Структура каталогов Linux имеет иерархическую (древовидную) организацию, в основании которой лежит корневой (/, не путать с домашним каталогом администратора — `/root`) каталог. В качестве подкаталогов его выступают:

- `/bin` — каталог для исполняемых (иначе называемых двоичными, или бинарными, `binary`) файлов общего назначения; здесь помещаются обо-

лочки командной строки, общие команды управления файлами и их архивации, традиционные текстовые редакторы типа vi, и т.д.; именно каталог /bin в первую очередь просматривается на предмет поиска введенной с клавиатуры команды;

- /boot, как явствует из названия, содержит файл образа ядра, с которого загружается система;
- /dev — каталог для файлов устройств;
- /etc — каталог для конфигурационных файлов общего пользования;
- /home включает в себя домашние каталоги пользователей, со всеми их программами, личными конфигурационными файлами (имеющими в сеансе данного пользователя предпочтение перед общими файлами конфигурации) и данными;
- /lib — каталог общесистемных библиотек (аналогов DLL в Windows);
- /mnt — каталог для монтирования сменных накопителей (вроде дискет) или временно подключаемых файловых систем (например, FAT-раздела диска);
- /proc — виртуальная файловая система для чтения информации о процессах;
- /root — домашний ($HOME) каталог для суперпользователя;
- /sbin содержит бинарные исполняемые файлы, используемые для системного администрирования;
- /tmp включает в себя всякого рода временные файлы; как правило, этот каталог автоматически очищается при перезагрузке или через некоторое время;
- /usr — каталог для прикладных пользовательских программ со всеми их компонентами — исполняемыми, конфигурационными и разделяемыми файлами (/usr/bin, /usr/etc и /usr/share, соответственно), библиотеками (/usr/lib) и т.д. Важный подкаталог /usr/local предназначен для программ, не входящих в дистрибутив стандартно, — в него, по умолчанию, инсталлируются компилируемые из исходных текстов приложения, включая исполняемые файлы (/usr/local/bin), документацию (/usr/local/share/doc, /usr/local/share/info, /usr/local/share/man), библиотеки (/usr/local/lib);
- /var — каталог для часто меняющихся файлов: всякого рода системных журналов, почтовых и принтерных спулингов и т.д.

Кроме того, в иерархии могут присутствовать и некоторые другие каталоги, например, lost+found — для нарушенных фрагментов файлов, выявленных при проверке диска, /opt — для опциональных компонентов.

6.4 Права доступа и прочие атрибуты файлов

Как уже говорилось, все файлы в файловой системе Linux характеризуются набором атрибутов. Важнейшие из них — атрибуты принадлежности файлов и атрибуты доступа к ним. Именно их восприятие психологически наиболее сложно для перехода на Linux, и поэтому им следует уделить особое внимание.

Атрибутов принадлежности файла — три. Во-первых, каждый файл имеет своего владельца (`owner`). Это, как правило (хотя и не обязательно), — пользователь, создавший его или скопировавший. Во-вторых, файл принадлежит группе пользователей (`group`) — одной из тех, в которые входит его владелец. И, наконец, все прочие пользователи (реальные и виртуальные, `other`), то есть не являющиеся ни владельцем файла, ни членами группы, к которой он приписан, также имеют некоторое отношение к данному файлу (и, соответственно, могут иметь некоторые права на него).

Атрибутов доступа — также три: право на чтение (`read`), право на изменение (`write`) и право на исполнение (`execute`). Причем права эти понимаются различно в зависимости от принадлежности файла к одному из типов, выделенных при их классификации.

Наиболее важно различие в атрибутах доступа к обычным (`regular`) файлам и каталогам (`directory`). Так, право чтения обычного файла означает возможность просмотра его с помощью команд типа `cat`, `more`, `less`, текстовых редакторов или специализированных прикладных пакетов. Кроме того, обладатель права на чтение может скопировать файл.

Право на изменение позволяет изменить содержание файла, но не удалить, переместить или переименовать его — для этих операций требуется право изменения не файла, а каталога, в который он входит (о чем будет сказано ниже). В то же время отсутствие права на изменение данного файла не мешает его копированию — ведь при этом содержание исходного файла не претерпевает никаких изменений, так как создается новый файл, наследующий атрибуты не источника, а пользователя, запустившего процесс копирования.

Право на исполнение имеет смысл только для файлов исполняемых (опять рекурсия или, если угодно, тавтология), то есть откомпилированных бинарных программ и сценариев оболочки. Бесполезно было бы устанавливать право на исполнение для текстового документа или растрового графического изображения.

В то же время именно это право отличает файл с листингом пользовательского сценария от сценария собственно.

В отношении каталогов смысл атрибутов доступа иной. Право на чтение каталога означает возможность вывода его содержания (например, командой `ls <имя каталога>`), а также копирования каталога (в том числе и со всем

его содержимым, если права доступа к последнему тому не противоречат). Однако права чтения для выполнения этих действий мало — необходимо еще право на исполнение (о чем ниже).

Право на запись для каталога — это возможность изменять его содержимое, то есть записывать в него файлы или удалять их.

Наконец, право на исполнения в отношении каталога означает возможность перехода в него (командой cd имя_каталога) и последующего просмотра содержимого. Так что право исполнения и право чтения для каталога тесно сопряжены друг с другом, и обычно следует предоставлять для каталога или оба права, или ни одного. Тем не менее, права эти — разные, и иногда это может быть использовано для разграничения доступа.

Права доступа к существующим файлам могут быть просмотрены с помощью команды ls с параметром -l (от long). Рассмотрим их на примере каталога из предыдущего раздела:

```
ls -l ~/aspbooks
-rw-rw-r--2 alv alv 72611 Июл 18 11:36 admin.txt
-rw-rw-r--1 alv alv 79326 Июн 24 16:51 install.txt
-rw-rw-r--1 alv alv 70578 Июл  4 09:41 qstart.txt
drwxr-x--x 3 alv alv 4096 Июл 16 13:11 ris_admin
drwxr-x--x 5 alv alv 4096 Июн 24 09:48 ris_install
```

В этом списке за права доступа отвечают значения первого поля, содержание которого составляет десять символов. Первый из них определяет тип файла (обычный — -, то есть дефис, d — каталог, l — символическая ссылка и т.д.), о чем уже говорилось.

Остальные девять символов разделяются на три равновеликие части. Первая (слева на право) определяет права доступа для владельца (owner), вторая — для группы владельцев (group), к которой файл приписан, и третья — для всех прочих (other). Порядок символов следующий — чтение (r, read), запись (w, write), исполнение (x, от eXecute). Наличие любого из символов в соответствующей позиции каждой части означает наличие данного права, знак дефиса — его отсутствие.

Так, в приведенном примере можно видеть, что для всех обычных файлов (опознаваемых по символу дефиса в первой позиции) его владелец имеет право на чтение и запись, но не исполнение (сочетание символов rw-), те же права присвоены и членам группы (четвертое поле записи). Все же остальные имеют только право на чтение файла (сочетание символов r--).

Иная картина будет для исполняемых файлов, например, пользовательских сценариев, что можно видеть на следующем примере ls -l bin/:

```
-rwxr-x--x 1 alv alv 39 Июл 1 09:34 oo
-rwxr-x--x 1 alv alv 35 Июн 30 09:34 so
```

Здесь владелец файлов обладает всей полнотой прав — чтения, записи и исполнения (rwx), члены группы — право на чтение и исполнение, но не изменение (r-x), прочие же могли бы запускать сценарии на исполнение без права измения, но поскольку просмотреть их не в состоянии (--x), то и исполнить тоже[2].

Вернемся, однако, к первому примеру и рассмотрим для него атрибуты доступа к каталогам:

```
drwxr-x--x 3 alv alv 4096 Июл 16 13:11 ris_admin
drwxr-x--x 5 alv alv 4096 Июн 24 09:48 ris_install
drwxr-x--x 11 alv alv 4096 Июл 13 17:19 ris_user
```

Как и в случае с файлами, владелец имеет все права в отношении этих каталогов — право просмотра их содержимого (r), право удалять или записывать в них файлы (w) и право перехода в каталог (x). Права членов группы уже, им разрешается просматривать каталоги и переходить в них (r-x). Наконец, за прочими есть только право исполнения, то есть перехода в каталог: ни изменить его содержание, ни даже просмотреть его они не в могут (--x).

Может показаться, что такой набор прав для пользователей лишен смысла: чтобы был толк от возможности перейти в каталог, следует иметь и право его просмотра. В данном примере это так и есть. Однако вернемся к примеру с пользовательскими сценариями. Если просмотреть права доступа к содержащему их каталогу, можно увидеть те же атрибуты доступа:

```
drwxr-x-x 2 alv alv 4096 Июл 1 09:34 bin/
```

то есть полный набор прав для доступа владельца, возможность чтения и перехода для членов группы и лишь возможность перехода — для всех остальных. То есть пользователь, не являющийся владельцем каталога и не входящий в его группу, может перейти в этот каталог и запустить на исполнение любой из содержащихся там сценариев — как мы помним, такое право в отношении их ему дано. Правда, при условии знания их точного имени — принцип дополнения команды клавишей `Tab` для него не сработает ввиду запрета на чтение содержимого каталога.

Приведенная форма записи прав доступа называется символьной. Она не является единственной — существует еще т.н. абсолютная, или цифровая, форма.

Просмотреть ее можно, например, с помощью команды stat с именем файла в качестве аргумента. Так, для приводимого выше в качестве примера файла admin.txt ответ на эту команду будет следующим:

```
File: "admin.txt"
```

[2] Таким образом атрибут --x в данном случае является бесполезным.

```
Size: 77240       Blocks: 160       Regular File
Access: (0664/-rw-rw-r--) Uid: (500/alv) Gid: (500/alv)
Device: 2103       Inode: 2327203     Links: 2
Access: Wed Jul 18 09:40:14 2002
Modify: Wed Jul 18 13:52:12 2002
Change: Wed Jul 18 13:52:12 2002
```

Некоторые из выведенных здесь атрибутов нам уже знакомы, о других речь пойдет дальше. Сейчас же остановимся только на поле Access, которое, собственно, и отражает атрибуты доступа. Второе его значение, после символа /, понятно — это права доступа в символьной форме (-rw-rw-r--). А первое значение (0664) и являет собой абсолютную форму нотации прав доступа.

Первая цифра этой записи (0), хотя и имеет отношение к правам доступа, рассматриваться пока не будет. Оставшиеся три (664) в точности соответствуют трем группам символов с символьной нотации: это права владельцев, группы владельцев и прочих.

Образуются эти цифры простым суммированием прав доступа для каждого из уровней принадлежности, поскольку в абсолютной нотации каждому из прав доступа соответствует цифра: то есть -rw-rw-r-- в символьной нотации — это 110 110 100 в двоичной системе исчисления, что в восьмеричном исчислении и дает 664.

Для исполняемого сценария из второго примера картина будет другой:

```
Access: (0771/-rwxrwx--x)
```

то есть пользователь и члены его группы имеют права чтения, записи и исполнения (111=7), а все прочие — лишь право исполнения (001=1).

Типичный же набор атрибутов доступа для каталога будет таким:

```
Access:(0755/drwxr-xr-x)
```

То есть право чтения, записи и исполнения для владельца 111=7, право чтения и исполнения 101=5 — для группы и прочих.

Легко подсчитать, что предоставление всех возможных прав доступа для владельца, группы владельцев и все остальных выразится значением 777 (111 в каждой позиции), а отсутствие любых прав для них — значением 000. Ниже будет показано, что в одних случаях удобнее пользоваться символьной нотацией, в других — абсолютной.

Теперь следует поговорить о том, откуда берутся атрибуты доступа и принадлежности. Они возникают в силу создания файлов пользователями (в широком

смысле слова, включая администратора и виртуальных пользователей). То есть файл, созданный пользователем alv, будет иметь его своим владельцем, и принадлежать к основной группе, в которую тот входит (о чем подробнее — в следующей главе), атрибуты owner и group для файла, созданного администратором, будут иметь значение root, и т.д.

Права доступа для файла определяются не правами пользователя, их создавшего, а правами запущенного этим пользователем процесса, породившего данные файлы — в общем случае, как говорилось в главе о процессах, они могут и не совпадать. По умолчанию каждый создаваемый файл получает атрибуты доступа, определяемые командой umask. Формат ее следующий:

```
umask 022
```

Аргумент команды и представляет собой маску прав доступа каждого вновь создаваемого файла. Значение цифр подобно таковым абсолютной нотации, но достигается не суммированием права чтения (4), изменения (2) и исполнения (1), а их вычитанием из цифры 7 (максимального значения прав в абсолютной нотации).

Аргумент команды umask в приведенном примере означает, что для каждого вновь создаваемого файла будут устанавливаться права чтения, записи и исполнения для его владельца (7-4-2-1=0), и права чтения и исполнения для группы и всех остальных (7-4-1=2).

Приведенное значение аргумента umask по умолчанию (022) в дистрибутиве **ASPLinux** определено глобально в файле /etc/init.d/functions. При необходимости его изменения соответствующее значение вносится в файлы конфигурации командной оболочки пользователя (для оболочки bash — обычно в файл /.bash_profile). Так, строка

```
umask 027
```

определяет, что по умолчанию любой создаваемый данным пользователем файл будет доступен владельцу для чтения, записи и исполнения, группе — только для чтения и исполнения, прочим же — недоступен вообще.

Впрочем, и атрибуты доступа, и атрибуты принадлежности файла не есть нечто неизменное. Владелец файла может легко сменить все права на доступ файла для самого себя, группы и прочих. Может он и назначить принадлежность файла другой группе, хотя и не любой, а только той, членом которой он сам является. Однако изменить владельца файла (то есть назначить владельцем файла другого пользователя) он не имеет права. Это — прерогатива исключительно администратора, который располагает полномочиями изменить для файла все атрибуты доступа и принадлежности (как, впрочем, и почти все прочие атрибуты).

Для изменения владельца файла предназначена команда chown (от «*change owner*»). Она вводится в форме

```
chown newowner file
```

где newowner — имя нового владельца файла file. Из сказанного выше ясно, что воспользоваться этой командой может только администратор. Пользователю же доступна команда для смены принадлежности группе chgrp (от change group):

```
chgrp newgroup file
```

где newgroup — имя новой группы, которой будет принадлежать файл file. Как уже говорилось, для выполнения этого действия владелец файла должен сам быть членом группы newgroup.

Изменить атрибуты принадлежности можно и одной командой chown в следующей форме:

```
chown newowner.newgroup file
```

Для смены атрибутов доступа служит команда chmod (от change mode), где в качестве аргумента используется имя файла, а параметры определяют присвоение (+) или отнятие (–) соответствующих прав: чтения (r), записи (w) и исполнения (x). Например, команда

```
chmod +rwx file
```

присвоит всем пользователям права на чтение, изменение и исполнение файла file, а команда

```
chmod -w file
```

отнимет у всех (включая владельца) право изменять этот файл. Однако права можно присваивать или отнимать и избирательно для разных уровней принадлежности: для владельца (u, от user), группы (g) и всех остальных (o — other). Для этого соответствующие символы (в любом наборе и сочетании) указываются слева от знака присвоения/отнятия: команда

```
chmod ug+wx file
```

присвоит право изменения и исполнения файла владельцу и его группе, а команда

```
chmod o-w file
```

отнимет право изменения файла всеми остальными. Кроме того, есть и форма

```
chmod a+rwx file
```

присваивающая права доступа всем уровням принадлежности, что эквивалентно ранее приведенной форме без указания принадлежности вообще.

Как и большинство команд Linux, все три команды для смены атрибутов могут использоваться рекурсивно, то есть применительно к каталогам, всем входящим в них подкаталогам и составляющим их файлам. Так, команда

```
chown newowner -R dir1/
```

сменит владельца не только каталога dir1, но и всех входящих в него подкаталогов и файлов. То же справедливо и для команд chgrp и chmod.

Именно при рекурсивном исполнении проявляется некоторое неудобство символьной нотации атрибутов доступа. Следует подчеркнуть, что команда chmod изменяет только те права доступа, и только для тех уровней принадлежности, которые явно указаны в виде ее параметров. То есть команда

```
chmod g-w -R dir1/
```

только отнимет право изменения каталога dir1 и всех его файлов у членов группы, не затронув прав доступа (которые в общем случае могут быть самыми разными для файлов и подкаталогов) владельца и остальных. Правда, есть и возможность «эксклюзивного» присвоения какого-либо права. Для этого используются параметры =r, =w, =x, которые устанавливают для аргумента (файла или каталога, в том числе и рекурсивно) только указанное право и никакое другое. Так, команда

```
chmod g=r -R dir1/
```

присвоит группе право чтения каталога dir1 и его составляющих, одновременно отнимая все остальные права — записи и выполнения, а также просмотра данного каталога. Аналогично

```
chmod a=x -R dir1/
```

для всех пользователей (владельца, группы и прочих) присвоит исключительно право выполнения для структуры dir1, отнимая право на запись и чтение.

Однако именно в случае рекурсивного выполнения команды chmod удобнее может оказаться абсолютная нотация атрибутов доступа, поскольку при ней все они определяются одной командой для всех уровней принадлежности. Особенно эффективно использование абсолютной нотации администратором, так как он может в один прием унифицировать политику доступа к файлам всех пользователей вообще.

Остановимся на атрибутах времени, для чего снова обратимся к команде stat. Три последние строки ее вывода имеют примерно следующий вид:

```
Access: Mon Jul 16 21:53:31 2001
Modify: Wed Jul  4 09:41:11 2001
Change: Wed Jul 18 18:28:01 2001
```

Можно видеть, что все три атрибута, имеющие отношение к существованию файла во времени — время доступа (Access Time), время модификации (Modification Time) и время изменения (Change Time) имеют разные значения. Характерно, что ни один из них не отражает время создания файла как такового:

- время доступа устанавливается при любом обращении к файлу — например, считыванию его прикладной программой;
- время модификации фиксируется в момент изменения содержания файла;
- время изменения — это время смены атрибутов файла, например, прав доступа.

Именно последний атрибут можно с определенной долей условности считать временем создания файла — в том случае, если за время его существования права доступа не изменялись.

Для изменения атрибутов времени файла применяется команда touch. Запущенная без параметров, с именем файла в качестве аргумента, она присваивает этому файлу атрибуты времени текущего момента. Если файл с таким именем не существует, он будет создан (пустым) этой командой.

С помощью параметров команды touch можно изменить время доступа (-a), модификации (-m), причем приписать им любое заданное время вместо текущего (-d), или заимствовать временные атрибуты у некоего файла (-f имя_файла).

Глава 7

Управление учетными записями пользователя

Понятие учетных записей пользователей и их групп — третье из ключевых понятий Linux. К его рассмотрению мы и перейдем в этой главе.

Следует сразу подчеркнуть, что понятие пользователя системы отнюдь не совпадает с пользователями ее в физическом смысле слова. Учетные записи могут существовать и для т.н. виртуальных пользователей, не соотносящихся ни с какими реальными лицами. И потому не очень изящное выражение «учетная запись пользователя» (account, бюджет) лучше отражает существо дела. Чтобы осознать это, достаточно посмотреть, как хранятся данные о пользователях.

Для этого предназначен файл /etc/passwd. Он представляет собой простой текстовый файл, содержащий многие десятки строк, каждая из которых представляет собой учетную запись одного пользователя. Одного взгляда на нее достаточно, чтобы понять, насколько мало отношения имеет учетная запись к пользователю как физическому лицу:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
...
alv:x:500:500:Alex:/home/alv:/bin/bash
```

Тем не менее, разберем отдельную запись на примере именно реального пользователя (последняя строка примера). Поля записи разделяются двоеточиями. Первое из них — имя пользователя (т.н. username, не вполне точно называемое также login).

Следующее поле — пароль (password). В некоторых дистрибутивах Linux оно действительно содержит односторонне зашифрованный пароль. Однако в **ASPLinux** это — лишь ссылка на файл с реальными зашифрованными паролями

(`/etc/shadow`, о котором разговор пойдет в главе о безопасности системы).

Следующие два поля — идентификаторы: первое — пользователя (`UID`, `User IDentificator`), второе — группы (`GID`, `Group IDentificator`), те самые, о которых вскользь упоминалось в главе о правах доступа. Именно `UID` однозначно определяет пользователя при его входе в систему, тогда как имя пользователя выступает только в качестве его синонима. Об идентификаторе группы же подробнее будет сказано ниже.

Далее — поле реального имени пользователя (`gecos`). Его формат произвольный, теоретически оно может содержать анкетные данные пользователя, если он является физическим лицом. Такие данные могут использоваться, например, почтовыми программами.

Последние два поля — это путь к домашнему каталогу пользователя (`home directory`) и к исполнимому файлу его командной оболочки по умолчанию (`shell`).

Такова структура записи для реального пользователя, с которой совпадает и учетная запись администратора. Не все из этих полей обязательны к заполнению: таковыми являются только `username`, `UID`, `GID` и `home directory`. Именно они и заполнены у т.н. «виртуальных» пользователей типа стартовых демонов. Отсутствие пароля, в принципе, допустимо и для реального пользователя, хотя это не рекомендуется из соображений безопасности.

Понятие группы пользователей дополняет понятие пользователя. Каждый из них входит как минимум в одну, основную, группу, хотя может быть членом нескольких других, дополнительных.

Данные о группах хранятся в файле `/etc/group`. Если рассмотреть его, можно видеть, что большинство групп — отнюдь не определение реальных пользователей:

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
...
alv:x:500:
```

Структурно файл `/etc/group` подобен файлу `/etc/passwd` и также образован серией записей, разделенных символом `:` (двоеточие) на поля. Первое поле — имя группы, однозначно связанное с ее идентификатором `GID` (`Group IDentificator`), занимающем третье поле (аналогично имени пользователя и его `UID`). Между ними, во втором поле — пароль группы, который обычно не используется и остается, почему поле это или пусто, или занято неким разрешенным символом.

Четвертое поле записи о группе — список входящих в нее пользователей. Как

уже говорилось, каждый из них входит, по крайней мере, в одну группу — именно ту, GID которой стоит у него в соответствующем поле учетной записи и имя которой по умолчанию совпадает с именем пользователя. В этом случае четвертое поле записи о группе может быть и пустым, как в последней строке приведенного примера. Если же в данную группу входят еще какие-либо пользователи, они должны быть перечислены здесь явным образом. Причем допускается использование как их имен (username), так и идентификаторов (UID).

Группы предназначены для дополнительного разграничения доступа к файлам — как в плане его расширения, так и ограничения. Первое используется чаще.

Типичный пример — доступ к пользовательским файлам. По умолчанию некий user1 не только не имеет доступа к файлам user2, но даже не может просмотреть его домашний каталог (/home/user2) или зайти в него, и наоборот. Так что обмен данными между ними невозможен.

Однако при необходимости такого обмена (а она возникает, например, при работе над единым проектом) выход есть. Для этого user1 достаточно сделать пользователя user2 членом своей группы (по умолчанию — user1) и присвоить своим файлам требуемые права доступа для ее членов (например, чтения и исполнения). Правда, для обратной связи user2 должен сделать user1 членом своей группы. Кроме того, по умолчанию в этом случае оба пользователя получают доступ (по крайней мере, для чтения и исполнения) ко всем файлам друг друга.

Поэтому может использоваться и другой способ, требующий вмешательства администратора. Он создает отдельную группу, не ассоциированную ни с одним из упомянутых пользователей, и делает их обоих ее членами. Те же должны каждый самостоятельно установить принадлежность к этой группе тех файлов и каталогов, которые требуются им для совместной работы, сохранив для прочих принадлежность к группе исходной.

Ограничение доступа на уровне группы обычно используется администратором и применяется в отношении не реальных, а виртуальных пользователей, таких, как http-, ftp- и почтовые клиенты, которые тоже должны иметь свои учетные записи. Впрочем, это относится уже к вопросам безопасности системы.

Создание учетных записей пользователей и групп осуществляется администратором. Первой цели служит команда useradd (или adduser, представляющей собой символическую на нее ссылку). Запущенная без параметров, в виде

```
useradd newuser
```

она создает учетную запись нового пользователя с именем newuser и его

домашним каталогом /home/newuser, в который копируются конфигурационные файлы командной оболочки (по умолчанию — bash). Файлы, подлежащие копированию, определяются содержимым каталога /etc/skel и могут быть отредактированы администратором в текстовом редакторе.

Пароль нового пользователя при этом не задается, и авторизоваться новый пользователь пока не может: предварительно администратор должен прибегнуть к команде

```
passwd newuser
```

которая запросит новый пароль, а затем его повторение.

Более гибкое управление учетными записями возможно благодаря многочисленным опциям команды useradd, с которыми можно ознакомиться на странице ее интерактивного руководства — man useradd или info useradd. Для создания новых групп используется команда groupadd.

Изменить содержание полей учетной записи пользователя можно с помощью команды chfn. Данная с именем пользователя в качестве аргумента, она последовательно запрашивает его полное имя, его служебные атрибуты (номер офиса и телефон), а также домашний телефон:

```
# chfn zus
Changing finger information for zus.
Name [Zsh User]:
Office [305]:
Office Phone [2308158]:
Home Phone [1926208]:
```

Старые значения этих полей даются в скобках, как ответ по умолчанию: если нет необходимости менять какое-либо из них, достаточно просто нажать Enter.

Существуют также многочисленные графические утилиты управления учетными записями пользователей и их групп. Одна из наиболее универсальных из них доступна в меню «*Приложения*», далее — «**Системные параметры**» — «**Пользователи и группы**».

Запуск этой программы приводит к выводу окна со строкой меню, инструментальными кнопками и двумя панелями (рис. 7.1): в левой выведен полный список существующих пользователей, в правой — их групп.

Действия, доступные через меню и через инструментальную панель, совершенно идентичны. Среди них: создание учетной записи нового пользователя, редактирование и удаление существующей, создание, редактирование и удаление записи для группы.

Создание новой учетной записи пользователя (через меню или кнопку инструментальной панели) начинается с ввода его имени (username), после чего в панели Свойства пользователя с тремя закладками определяются ее поля.

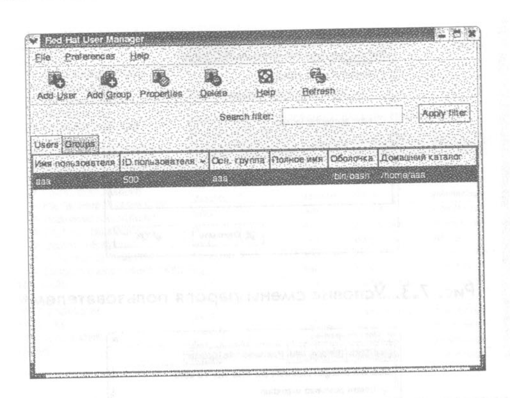


Рис. 7.1: Графическая утилита управления пользователями — **system-config-users**

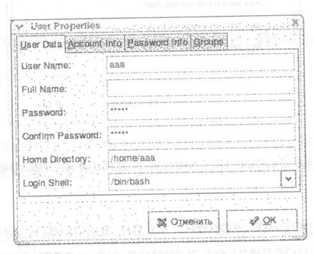


Рис. 7.2: Заполнение основных полей учетной записи пользователя

В закладке «*User Data*» (рис. 7.2) можно скорректировать введенное имя, установить командную оболочку по умолчанию (любую из доступных в системе, то есть перечисленных в файле `/etc/shells`) и домашний каталог (по умолчанию — `/home/username`).

В закладках «*Password Info*» (рис. 7.4) и «*Account Info*» (рис. 7.3) определяются условия смены пароля. Все они касаются времени смены пароля и истечения срока его действия.

Здесь можно, воспользовавшись соответствующими переключателями, установить срок истечения действия пароля, дату уведомления об истечении этого срока, а также истечение срока доступа данного пользователя вообще.

В закладке «*Groups*» (рис. 7.5) устанавливаются: основная группа для данного пользователя, а также список групп, к которым он приписан.

Для редактирования существующей учетной записи некоего пользователя требуется зафиксировать на ней курсор и кнопкой «**Properties**» на инструментальной панели или через меню («*User*»- «*Properties*») вызвать ту же самую панель, что и при создании пользователя, в которой и меняется, при необхо-

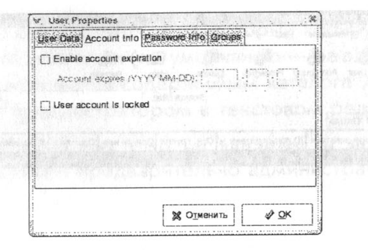


Рис. 7.3: Условия смены пароля пользователем

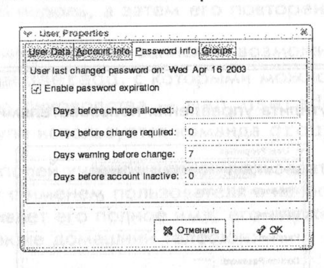


Рис. 7.4: Условия смены пароля пользователем

димости, содержание любых полей. Если выбрать кнопку (или пункт меню) удаления пользователя, последует запрос на подтверждение действия.

Управление группами осуществляется аналогично. Для создания группы выбирается пункт меню «*Groups*»- «*Add Group*» и вводится имя новой группы. Далее, после обращения к пункту «*Groups*»- «*Properties*», вызывается панель «*Properties*» (рис. 7.5), через который в группу включаются или исключаются пользователи.

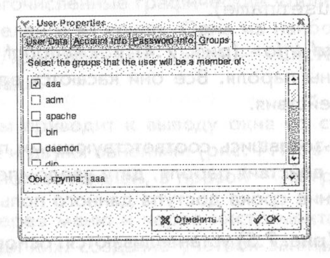


Рис. 7.5: Изменение списка пользователей, входящих в группу

На удаление группы, как и пользователя, запрашивается подтверждение. Напомним, что все действия в программе **system-config-users** осуществляются только от лица администратора.

Рис. 7.6: Управление учетными записями пользователей

7.1 Управление учетными записями пользователей при помощи Webmin

Webmin позволяет управлять пользователями и группами пользователей при помощи модуля «*Пользователи и группы*», который находится в разделе «*Система*».

На первой странице модуля показаны список с пользователями системы и список групп пользователей. Также можно посмотреть список пользователей, которые в данный момент работают в системе. Для его формирования следует нажать на кнопку «*Пользователи, вошедшие в систему*», которая находится в самом низу страницы. Там же есть кнопка, формирующая список, в котором показывается история входов пользователей в систему. Для формирования списка входа только одного пользователя, логин этого пользователя необходимо ввести в соответствующем поле или выбрать его в окне, появляющемся после нажатия на кнопку «...».

Для добавления нового пользователя при помощи Webmin, необходимо выбрать ссылку «*Создать нового пользователя*» и в появившейся странице заполнить необходимые поля. Специфичным для Webmin является пункт «*Создать пользователя в других модулях?*». Если этот пункт включен, при добавлении нового пользователя будут вноситься изменения и в другие модули. Например, если в модуле работы с квотами файловых систем для вновь создаваемых пользователей были установлены квоты по умолчанию, эти квоты будут применены для созданного пользователя. Другой пример — это модуль для работы с **Samba**-сервером. При создании нового пользователя он будет добавлен в список пользователей **Samba**.

Если в главном окне списка выбрать ссылку с именем пользователя, появится страница, аналогичная странице добавления нового пользователя. При ее

помощи можно изменить параметры учетной записи выбранного пользователя. Также в этом окне присутствуют дополнительные кнопки: «**Прочитать почту**» и «**Удалить**». Поскольку Webmin запускается с правами суперпользователя root, у администратора появляется возможность чтения почты всех пользователей системы. На самом деле, для работы с почтой будет вызван другой модуль — «*Конфигурация sendmail*». Кнопка «**Удалить**» — удаляет учетную запись текущего пользователя.

Для добавления новой группы, выберите ссылку «*Создать новую группу*». В появившемся окне в поле «*Имя группы*» введите имя создаваемой группы. Поле «*ID группы*» Webmin обычно заполняет сам, там будет введен следующий свободный ID, который выбирается в диапазоне от 500 до 60000. Пароль группы устанавливается только тогда, когда необходимо передать управление группой другому пользователю. К сожалению, в Webmin не реализована возможность управления группой другими пользователями, поэтому пароль на группу устанавливать не обязательно. В списке «*Члены группы*» должны быть перечислены все пользователи, входящие в данную группу. Для того, чтобы не допустить ошибку при вводе имен пользователей, входящих в группу, воспользуйтесь кнопкой «...». В появившемся окне выберите необходимых пользователей и нажмите кнопку «**OK**». После ввода необходимых параметров, для создания группы нажмите на кнопке «**Создать**».

Для редактирования группы пользователей на главной странице модуля необходимо выбрать имя группы. Страница редактирования свойств группы похожа на страницу добавления новой группы. Если при редактировании параметров у группы был изменен ID, желательно в разделе «*Сменить ID группы для файлов*», выбрать пункт «*Все файлы*». В этом случае у всех файлов в файловой системе, которые принадлежат данной группе, старый ID будет изменен на новый. Если не выбрать «*Все файлы*», то файлы будут принадлежать группе со старым ID и члены редактируемой группы не получат доступа к этим файлам на основе прав группы. Для удаления группы воспользуйтесь кнопкой «**Удалить**».

Глава 8

Настройка консольного режима

Текстовая консоль Linux обладает двумя важнейшими свойствами — поддержкой виртуальных консолей (каждая из которых ведет себя как соответствующее физическое устройство) и экранного буфера. Кроме того, она позволяет подгружать экранные шрифты и раскладки клавиатуры, отличные от принятых по умолчанию, а также использовать нестандартные экранные разрешения.

Прежде чем описывать настройку консоли, нужно дать понятие о уровнях запуска ядра. Для них определено шесть значений, которые можно увидеть в конфигурационном файле /etc/inittab, который считывается первым в ходе загрузки системы:

```
#   0 - останов системы
#   1 - однопользовательский режим
#   2 - многопользовательский режим без поддержки сети
#   3 - полный многопользовательский режим
#   4 - не используется
#   5 - полный многопользовательский режим с возможностью запуска
#       системы X Window
#   6 - перезагрузка системы
```

Каждый уровень запуска — это некий список сервисов, которые стартуют автоматически. Назначение их — привести систему в какое-то предопределенное состояние.

По умолчанию в **ASPLinux** предусмотрено шесть виртуальных консолей. Количество это определяется в файле /etc/inittab следующими строками:

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Каждой виртуальной консоли соответствует строка, поля в которой, разделенные знаком :, определяют ее свойства.

Первое поле в каждой строке — идентификатор консоли, соответствующий номеру терминала как устройства в каталоге /dev (/dev/tty1, /dev/tty2 и т.д.).

Второе поле — уровни запуска ядра системы, при которых данная консоль доступна (со второго, то есть многопользовательского режима без поддержки сети, по пятый, запуск X Window System).

Поле «*respawn*» предписывает запуск процесса, указанного в последнем поле — то есть активизацию терминала процессом mingetty, и перезапуск после его окончания. Это обеспечивает вывод на экран приглашения к авторизации по завершении сеанса пользователя (командой exit или logout).

Для уменьшения количества консолей достаточно удалить нужное количество строк, их описывающих: это делают для высвобождения некоторого (очень незначительного) количества памяти и имеет смысл только на компьютерах с очень ограниченными ресурсами.

Увеличение количества консолей достигается добавлением записей с номерами 7, 8, и т.д. и соответствующими им номерами терминалов в качестве аргументов команды mingetty — 7, 8 и т.д.

Переключение в консоль осуществляется комбинацией клавиш Alt+F#, с 1-й по 12-ю с левым Alt, с 13-й по 24-ю — с правым Alt. Кроме того, комбинацией Alt+Right можно циклически перемещаться в следующую активизированную консоль, комбинацией Alt+Left — в предыдущую.

За активизацию графической консоли отвечает строка

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

которая предписывает запуск X Window System на первой свободной консоли. То есть при активизации шести текстовых консолей X Window System будет автоматически запущена на седьмой консоли, при восьми — на девятой, и т.д. Второй же запущенный сеанс X Window System оккупирует следующую свободную консоль (то есть восьмую или, соответственно, десятую).

Прочие свойства консоли Linux, такие как ее видеорежим, экранные шрифты и раскладки клавиатуры, определяются следующими по порядку считывания стартовыми файлами — /etc/rc.d/rc, /etc/rc.d/rc.sysinit и /etc/rc.d/rc.local.local, называемыми некоторыми пользователями ресурсными файлами (rc– от Resouces Configuration).

Стандартное разрешение текстовой консоли Linux — 25 строк на 80 колонок (обозначаемое обычно как 80x25), что соответствует растровому экранному шрифту 8x16. Однако в текстовом режиме доступны и другие разрешения —

80x28, 80x30 и т.д. Для переключения между ними используется команда resizecons, в которой в качестве параметров можно задать матрицу разрешения в формате

```
resizecons 80x28
```

или

```
resizecons 80 28
```

а также просто количество строк

```
resizecons -l 30
```

Чтобы выбранный видеорежим загружался по умолчанию, эту команду следует внести в один из стартовых rc-файлов (наиболее подходящий из них — /etc/rc.d/rc.local.local).

Следует только учесть, что загружаемый в **ASPLinux** по умолчанию кириллический шрифт для текстового режима — UniCyr_8x16, — рассчитан именно на стандартное разрешение, и при смене его русские буквы в текстах исчезнут, сменившись соответствующими им символами верхней части латинской кодовой таблицы (обычно символами псевдографики).

Чтобы этого не произошло, следует предусмотреть загрузку кириллических шрифтов, рассчитанных на другие разрешения. Файлы экранных шрифтов текстового режима расположены в каталоге /lib/kbd/consolefonts. Для большинства языков и наборов символов доступны комплекты из трех шрифтов с матрицами 8x16, 8x14 и 8x8, что отражено в именах файлов, например, UniCyr-lenta-8x16.psf.gz, UniCyr_8x14.psf.gz и UniCyr_8x8.psf.gz для кириллических шрифтов в кодировке Unicode.

Настройки гарнитуры и кодировки консольного шрифта указываются в файле /etc/sysconfig/i18n[1]:

```
LANG="ru_RU.CP1251"
SYSFONT="UniCyr_8x16"
SYSFONTACM="cp1251"
```

Так, для кириллицы в кодировке Unicode доступны гарнитуры UniCyr-lenta-8x16 и UniCyr-sans-8x16. Последний предпочтителен для людей с плохим зрением.

[1] i18n - от англ. internationalization - интернационализация. Сокращение построено по следующему принципу: i в начале, n в конце и 18 букв между ними.

Кроме шрифтов Unicode, можно использовать и шрифты в иных кириллических кодировках, например, в традиционной для UNIX KOI8-R, кодировке CP866 или CP1251. Однако это может потребовать загрузки т.н. таблиц перекодировки.

Процесс этот тут не рассматривается, так как принятые в **ASPLinux** по умолчанию шрифты Unicode полностью снимают эту проблему.

Впрочем, к редактированию общесистемных файлов конфигурации (к коим принадлежит и /etc/sysconfig/i18n) следует прибегать только при необходимости. Поскольку изменить гарнитуру консольного шрифта можно и более простым способом — командой setfont. В качестве ее аргумента следует указать имя файла подходящей гарнитуры из указанного выше каталога. Например, команда

```
setfont /lib/kbd/consolefonts/UniCyr-lenta-8x16
```

установит шрифт UniCyr-lenta-8x16. Того же эффекта можно добиться и командой consolechars с опцией -f и аргументом в виде имени файла шрифта

```
consolechars -f /lib/kbd/consolefonts/UniCyr-lenta-8x16
```

что по ряду соображений предпочтительней. Следует помнить только, что в любом случае экранный шрифт изменится для всех виртуальных консолей одновременно.

Параметры раскладки клавиатуры прописаны в файле /etc/sysconfig/keyboard.

```
KEYTABLE="имя_файла_раскладки"
```

Доступные значения KEYTABLE определяются набором файлов в каталоге /lib/kbd/keymaps/i386/qwerty. В частности, среди кириллических раскладок присутствуют варианты с расположением клавиш как в MS DOS или Windows, и с самыми разнообразными переключателями. Так, если внести в эту строку значение

```
KEYTABLE="ruwin_cplk"
```

в результате получится кириллическая раскладка с Windows-расположением клавиш и переключением с латиницы на кириллицу посредством клавиши Caps Lock (прежняя ее функция — перевод в верхний регистр, — будет при этом достигаться одновременным нажатием Shift+Caps Lock).

Следует помнить, что, в отличие от настройки клавиатуры через **Панель управления** (как это было описано в *«Руководстве по установке»*), редактирование файла /etc/sysconfig/keyboard скажется только на текстовом режиме: раскладка и переключатель клавиатуры в X Window System останутся неизменными (об этом будет рассказано в следующей главе).

Глава 9

Настройка X Window System

В большинстве случаев настройка X Window System корректно осуществляется на стадии установки **ASPLinux**. Однако в ряде случаев возникает необходимость ручной корректировки — в случае неправильного определения параметров монитора, видеокарты, желания использовать нестандартные раскладки клавиатуры и их переключатели, а также при смене оборудования.

9.1 Настройка с помощью программы system-config-display

Для настройки системы `X.Org`, входящей в состав дистрибутива **ASPLinux**, используется программа **system-config-display**.

После начала работы, программа автоматически запускает X Window System, самостоятельно определяет тип видеокарты и монитора и предлагает (рис. 9.1).

Более подробные сведения о найденном оборудовании можно получить, перейдя по вкладке «**Оборудование**» (рис. 9.2).

Если оборудование было определено неверно - необходимо нажать кнопку «**Настроить**» и выбрать тип монитора и видеокарты вручную.

Результатом работы программы `system-config-display` является создание (в каталоге `/etc/X11`) файла `xorg.conf`.

Ниже будет рассмотрена структура файла `xorg.conf`.

9.2 Структура конфигурационного файла xorg.conf

Необходимость в ручной правке конфигурационного файла `/etc/X11/xorg.conf` возникает при использовании нестандартных раскла-



Рис. 9.1: Программа system-config-display: выбор разрешения и глубины цвета

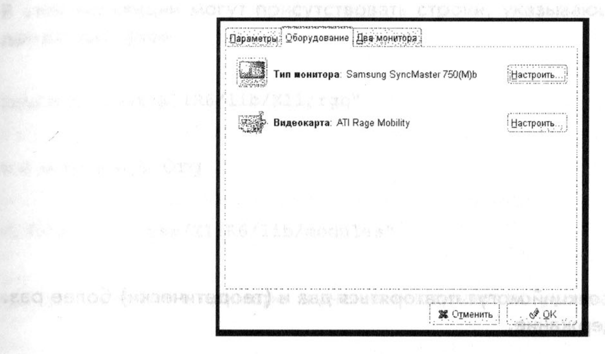


Рис. 9.2: Программа system-config-display: подробные сведения о найденном оборудовании

док клавиатуры и их переключателей, собственных видеорежимов и в ряде других случаев.

Для внесения правки в файл xorg.conf следует представлять себе его структуру. Она образована рядом секций, каждая из которых начинается строкой вида

```
Section "Имя_секции"
```

и заканчивается строкой

```
EndSection
```

Секции файла xorg.conf имеют следующие имена:



Рис. 9.3: Программа `system-config-display`: выбор модели монитора из списка

- ServerLayout,
- Files,
- Module,
- InputDevice,
- Monitor,
- Device,
- Screen.

Некоторые их этих секций могут повторяться два и (теоретически) более раз. Рассмотрим их содержание.

9.3 Секция ServerLayout

Секция `ServerLayout` описывает конфигурацию X-сервера и устройств (видеосистемы, клавиатуры, мыши) примерно в следующем виде:

```
Identifier      "Default Layout"
Screen       0  "Screen0" 0 0
InputDevice     "Mouse0" "CorePointer"
InputDevice     "Keyboard0" "CoreKeyboard"
```

Строка `Identifier` является просто идентификатором, единственное требование к которому — уникальность среди других секций `ServerLayout`. Это относится и к значениям этого параметра во всех остальных секциях, где он встречается.

Остальные строки указывают на идентификаторы конкретных секций `Screen` и `InputDevice` (см. ниже), задействованные в этой конфигурации.

9.4 Секция Files

Секция Files включает, как минимум, строку

```
FontPath "unix/:7100"
```

указывающую, что для управления шрифтами используется сервер шрифтов xfs. Значение это образуется из имени машины, на которой установлен сервер шрифтов, и порта. Значение из примера соответствует произвольной UNIX-машине, если xfs используется с удаленного компьютера, здесь следует указать его реальное сетевое имя. Собственно список доступных шрифтов находится в конфигурационном файле xfs, /etc/X11/fs/config.

В этой же секции могут присутствовать строки, указывающие на пути к базе данных шрифтов

```
RgbPath "/usr/X11R6/lib/X11/rgb"
```

и к модулям X.Org

```
ModulePath "/usr/X11R6/lib/modules"
```

где приведенные в примере значения выступают по умолчанию.

9.5 Секция Module

Секция Module указывает, какие модули должны быть загружены при старте X-сервера.

```
Load "GLcore"
```

9.6 Секция InputDevice

Эта секция повторяется минимум дважды. Первый раз она содержит описание клавиатуры, начинающееся строками ее идентификации и указания на драйвер:

```
Identifier "Keyboard0"
Driver     "keyboard"
```

Идентификатор должен быть уникальным среди всех секций InputDevice.

Далее следует описание геометрии клавиатуры

```
Option "XkbModel" "pc105"
```

набора символов и конкретной раскладки

```
Option "XkbLayout" "us,ru(winkeys)"
```

Вслед за этим указывается переключатель клавиатурных раскладок, например

```
Option "XkbOptions" "grp:ctrl_shift_toggle,grp_led:scroll"
```

обеспечивающий переключение с латиницы на кириллицу с помощью комбинации клавиш Ctrl+Shift.

Использование группы grp_led:scroll не обязательно — она просто указывает, что переключение на кириллицу должно сопровождаться включением индикатора ScrollLock.

Следующая секция InputDevice посвящена описанию указателю мыши.

Как и предыдущая, она начинается с идентификатора устройства

```
Identifier  "Mouse0"
```

Она включает описание драйвера устройства, его имени (где /dev/mouse — символическая ссылка на реальное устройство, например, /dev/psaux для мышей PS/2) и протокола

```
Driver "mouse"
Option "Device"   "/dev/input/mice"
Option "Protocol" "IMPS/2"
```

Далее указывается необходимость эмуляции средней клавиши мыши одновременным нажатием крайних (для двухкнопочных моделей)

```
Option "Emulate3Buttons" "yes"
```

Очевидно, что для трехкнопочных моделей такой необходимости нет. Эмуляцию третьей кнопки следует отключить и для двухкнопочных мышей с колесом прокрутки, типа Microsoft Intellimouse или Genius NetScroll — в этом случае нажатие на колесо будет эквивалентно нажатию средней клавиши. Кроме того, для скроллирующих мышей следует дополнительно вписать строку

```
Option "ZAxisMapping" "4 5"
```

благодаря чему вращение колесика вперед и назад будет эмулировать клавиши PageDown и PageUp, соответственно.

Можно создать еще несколько секций InputDevice, например, для графического планшета. Нужно только следить, чтобы каждая такая секция имела уникальное значение поля Identifier. Выглядеть подобная секция может примерно так:

```
Section "InputDevice"
  Identifier "wacom"
  Driver "wacom"
  Option "Device"   "/dev/ttyS1"
  Option "AlwaysCore" "On"
  ...
EndSection
```

9.7 Секция Monitor

Подобно секциям для устройств ввода, секций, описывающих монитор, также может быть две и более, так как поддержка второго монитора возможна при использовании некоторых видеокарт (например, Matrox G-400/450) или просто двух видеокарт (например, для шин AGP и PCI).

Первые три строки любой из секций Monitor — идентификатор, имя производителя и название модели, — представляют собой произвольные символьные последовательности, лишь на первую накладывается требование уникальности (среди секций этого имени):

```
Identifier "Acer 76i"
VendorName "Unknown"
ModelName  "Unknown"
```

Две следующие строки присутствуют в секции обязательно — это частоты горизонтальной (в килогерцах) и вертикальной (в герцах) синхронизации:

```
HorizSync 30.0-64.0
VertRefresh 50.0-110.0
```

Значения обоих характеристик могут быть указаны в виде диапазона (как в примере), как список диапазонов или дискретных значений, разделенных запятыми. В качестве единиц измерения могут использоваться герцы или мегагерцы в первом случае, килогерцы и мегагерцы — во втором, если указать их явным образом (Hz, KHz, MHz) в конце строки. Опция


```
Option "dpms"
```

включает поддержку управления питанием для монитора (включение режимов Standby, Suspend, Off, о которых будет сказано ниже). Если необходимости в этом нет, строку эту можно удалить или закомментировать.

В секции `Monitor` могут присутствовать еще ряд дополнительных опций, например, для гамма-коррекции (`Gamma`) или для определения собственных видеорежимов (`UserModes`, `Mode` или `Modeline`), не совпадающих со стандартными VESA-режимами.

9.8 Секция `Device`

В этой секции описывается видеокарта. В простейшем случае она может состоять из двух строк — идентификатора и указания драйвера:

```
Identifier "Matrox Millennium G450"
Driver "mga"
```

Его для используемой модели следует подобрать в специальном каталоге `/usr/X11R6/lib/modules/drivers/`. В дистрибутив **ASPLinux** включены драйверы для всех современных моделей известных производителей — ATI, Matrox, NVidia, S3, 3Dfx, поддерживаются также встроенные видеосистемы i810/815/845/855. Кроме того, в последнее время некоторые производители (ATI, NVidia) разрабатывают собственные драйвера для использования в Linux. Они доступны на сайтах производителя.

Иногда в явном виде требуется указать объем видеопамяти (в Кбайт)

```
VideoRam "16384"
```

хотя обычно он определяется автоматически. Кроме того, ряд параметров видеокарты для современных моделей определяется втоматически их драйверами, но для более старых карт может потребоваться их указание в явном виде. В таком случае в секции `Device` могут присутствовать строки `Chipset`, `Ramdac`, `DacSpeed`, `Clocks`, `ClockChip` и другие, возможные значения которых следует искать в документации к конкретному драйверу.

Наконец, при использовании двух и более мониторов может потребоваться еще два параметра. Если мониторы подключены к различным видеокартам, для каждой из них должна быть создана своя секция `Device`, в которую вносится строка

```
BusID "PCI:1:0:0"
```

для карты с шиной AGP, и

```
BusID "PCI:шина:устройство:функция"
```

для PCI-карт.

При использовании видеокарты, поддерживающей работу с двумя мониторами (например, Matrox G-400/450), также создается две секции Device (на каждый монитор), в них добавляются строки

```
Screen "0"
```

для первого монитора, и

```
Screen "1"
```

для второго.

9.9 Секция Screen

Эта секция также может присутствовать в нескольких экземплярах. Она устанавливает соответствие между видео-картами, охарактеризованными в секциях Device, и мониторами, описанными секциями Monitor.

Каждая секция начинается строкой идентификации

```
Identifier "Screen#"
```

где Screen# принимает значение от 0 (для первой секции Screen) и выше (для последующих). Далее идут ссылки на идентификаторы задействованных в данной секции секций Device и Monitor

```
Device "Matrox Millennium G450"
Monitor "Acer 76i"
```

Соответственно для каждой секции Screen один (по крайней мере) из этих параметров должен отличаться от его значений во всех других секциях. Например, секция Screen, использующая кадровый буфер, примет вид:

```
Identifier "Screen1"
Device "Linux Frame Buffer"
Monitor "Acer 76i"
```

Следующая строка определяет глубину цвета, используемую по умолчанию при старте X-сервера:

```
DefaultDepth 24
```

Если эта строка отсутствует, X Window System будет загружена в режиме 8-битного цвета. Для секции, в которой как Device указан линейный кадровый буфер (Linux Frame Buffer), может потребоваться строка DefaultFbBpp с указанием собственной глубины цвета по умолчанию.

После этого в секцию Screen вводится одна или более субсекций Display, каждая из которых определяет набор разрешений для конкретной глубины цвета, обязательно завершаясь строкой EndSubSection. Например:

```
Subsection "Display"
Depth 24
Modes "1280x1024" "1154x852" "1024x768"
EndSubSection
```

Значение разрешения, указанное первым, будет применяться по умолчанию при данной глубине цвета. Переключаться между разрешениями в сеансе X Window System можно на лету — комбинацией клавиш Ctrl+Alt+Grey+ (повышение разрешения) и Ctrl+Alt+Grey- (его понижение), где Grey+ и Grey- — символы + и — на малой цифровой клавиатуре, соответственно.

Кроме реальных разрешений экрана, в каждой субсекции можно определить и т.н. разрешения виртуального экрана, превышающие реальные (Virtual x y, где размер по оси x в пикселях должен быть кратен 8 или 16), и начальные координаты верхнего левого угла видимой части виртуального экрана (ViewPort x y).

Последней может быть секция DRI (Direct rendering infrastructure), определяющая, какие пользователи могут работать с расширенной поддержкой трехмерной графики.

Строка

```
Mode 0666
```

разрешает это всем пользователям, в форме же

```
Mode 0660
```

это доступно только пользователям, включенным в группу, указанную в следующей строке:

Group"имя_группы"

Секция эта имеет смысл только в том случае, если ядро системы скомпилировано с поддержкой опции Direct Rendering Manager (см. ниже, в разделе 10.3).

9.10 Секция ServerFlags

Кроме этого, для ряда специальных настроек X Window System задействуется секция ServerFlags, помещаемая в начале файла xorg.conf (обычно после секции Files). Она может включать многочисленные опции, принимающие в большинстве случаев булевы значения — yes или no, on или off, true или false и обычно приводящие к запрещению каких-либо разрешенных по умолчанию действий. Среди них наиболее употребимы:

- Опция «DontZap», включение которой не позволяет прервать сессию графического режима комбинацией клавиш Ctrl+Alt+Backspace.
- Опция «DontZoom», которая при включении запрещает переключение разрешений экрана с помощью стандартных комбинаций клавиш Ctrl+Alt+Grey+/Ctrl+Alt+Grey-.
- Опция «VTSysReq» меняет способ переключения на другие виртуальные консоли из графического режима на комбинацию Alt+SysRq -> F# (вместо обычной комбинации Ctrl+Alt+F#), которая может быть задействована под внутренние нужды X Window System.
- Опция «NoPM» отключает все режимы энергосбережения.

  Если энергосбережение не выключено, и в секции Monitor включена опция «dpms», в секции ServerFlags можно задействовать различные режимы управления монитором с помощью строк, значения которых задаются в секундах:
- Опция «BlankTime» устанавливает время погасания экрана,
- Опция «StandbyTime», «SuspendTime» определяют время перехода в режимы ожидания и спящий, соответственно,
- Опция «OffTime» позволяет отключить питание монитора через заданный промежуток времени.

9.11 Секция Modes

Наконец, для определения собственных, нестандартных режимов может быть создана отдельная секция Modes (или, при необходимости, несколько таких

секций). Как и прочие секции, каждая из них должна включать уникальный идентификатор и набор опций, определяемых группой строк Mode, выступающей в качестве подсекции, заканчивающейся строкой EndMode. Они определяют пиксельную частоту режима, горизонтальную и вертикальную синхронизацию, сдвиги сигналов, и т.д.

Глава 10

Установка и обновление программного обеспечения

В дистрибутив **ASPLinux** включен ассортимент утилит и приложений, достаточный для решения широкого круга повседневных задач. Однако далеко не всегда при установке системы удается принять оптимальное решение по их выбору: практически неизбежно удаление ненужных программ и дополнительная установка необходимых.

Кроме того, программное обеспечение под Linux находится в непрерывном развитии. Постоянно выходят новые версии, функционально расширенные и (или) содержащие исправления ошибок. Появляются и совершенно новые программы, реализующие недоступные ранее функции.

Все это делает установку и обновление программного обеспечения повседневной задачей при администрировании системы. Задача эта разделяется на две части:

- установка и удаление программ из дистрибутива **ASPLinux**;
- установка программ из других источников.

Программы для Linux распространяются в двух видах: как откомпилированные бинарные пакеты и как пакеты с исходными текстами, требующими компиляции. Большинство программ, входящих в состав дистрибутива **ASPLinux**, представлены, в соответствие с условиями лицензии GPL, в обоих вариантах. Однако для установки штатного программного обеспечения используются, как правило, бинарные пакеты. Необходимость компиляции из исходных текстов возникает достаточно редко, за исключением перекомпиляции ядра, о чем пойдет речь в следующей главе.

Иное дело дополнительное (или обновленное) программное обеспечение, не входящее в состав дистрибутива. Большинство таких программ также доступно

как бинарные пакеты в форме, пригодной для установки в **ASPLinux**. Однако пакеты, откомпилированные для одного дистрибутива, на практике иногда не могут устанавливаться или использоваться в каком-либо другом без дополнительных операций. Кроме того, многие программы, особенно новые и находящиеся в стадии разработки, а также узкоспециализированные приложения, доступны только в виде исходных текстов. Поэтому ниже будут рассмотрены оба способа установки программ.

10.1 Представление о пакетах `rpm`

Дистрибутив **ASPLinux** унаследовал от своего предка — RedHat, — формат пакетов `rpm` (`RPM Package Manager`) — один из самых распространенных для откомпилированных программ для Linux. На дистрибутивных CD они расположены в каталоге `/mnt/cdrom/ASPLinux/RPMS/` (`/mnt/cdrom` — точка монтирования CD).

Просмотрев его содержимое, можно (с помощью команды `ls`) увидеть картину, подобную следующей:

```
CORBA-ORBit-0.4.2-0hlx1.1.asp.i386.rpm
ElectricFence-2.2.2-4.i386.rpm
GConf-1.0.0-1.asp.i386.rpm
Glide3-20001220-2.i386.rpm
Gtk-Perl-0.7003-0hlx1.1.asp.i386.rpm
ImageMagick-5.2.7-1.asp.i386.rpm
```

и так далее (общее число пакетов в **ASPLinux** превышает 1000). Каждый из перечисленных в списке файлов представляет собой отдельный пакет. Имя его несет в себе следующую информацию:

- название пакета (например, `ImageMagick`),
- номер версии (5.2.7-1),
- автор сборки пакета (компонент asp указывает, что пакет был собран специально для дистрибутива **ASPLinux**),
- архитектура компьютера, для которой предназначен пакет (`i386`),
- расширение `rpm`, указывающее на тип пакета.

Вместо указания на архитектуру может стоять компонент `noarch`, свидетельствующий, что пакет является кросс-платформенным, или `src`, обозначающий исходные тексты.

Для управления пакетами `rpm` обычно используется одноименная программа консольного режима, запускаемая из командной строки.

10.2 Управление бинарными пакетами с помощью программы rpm

Основное средство управления rpm-пакетами — консольная программа rpm. Запускается она следующим образом:

```
rpm -[основная опция][дополнительные опции] название_пакетов.rpm
```

Имена пакетов должны набираться полностью, включая номер версии, сведения об архитектуре и сборке, для чего следует воспользоваться возможностью автодополнения. Одновременно для обработки может быть указано сколько угодно имен пакетов. Представление же об опциях команды можно получить, запустив rpm без опций и аргументов.

К основным опциям, связанным с управлением бинарными пакетами, относятся:

- установка (-i), обновление (-U) или замена (-F --freshen, буквально «освежение») пакета,
- запрос информации о пакете (-q),
- удаление пакета (-e).

Различия внутри группы опций установки в том, что при собственно установке (-i) инсталлируется только отсутствующий пакет (в противном случае будет выдано сообщение, что пакет с данным именем уже установлен), при замене (-F) произойдет замена всех файлов старой версии пакета на более новую, а при обновлении (-U) совпадающие файлы старой и новой версий будут переписаны, недостающие файлы из новой — установлены, лишние файлы старой версии — удалены, за исключением всякого рода конфигурационной информации.

Дополнительно к одной из основных опций установки могут указываться (без пробела и разделяющего дефиса) опции -v (вывод текстовой информации) и -h (индикация процесса установки последовательными знаками #). Кроме того, есть еще группа дополнительных опций установки, отделяемых от основных пробелом и двойным символом дефиса (--). Это:

- --oldpackage, позволяющая заменить новый пакет на более старый при обновлении;
- --replacefiles, устанавливающая пакеты, даже если они перепишут файлы из уже установленных пакетов;

- --replacepkgs, устанавливающая пакеты, даже если некоторые из них уже установлены в системе;
- --force, принудительно устанавливающая пакеты и эквивалентная комбинации опций --replacepkgs, --replacefiles и --oldpackage;
- --nodeps, запрещающая проверку зависимостей перед установкой или обновлением пакета.

Таким образом, обычной формой использования rpm будут команды

```
rpm -ihv имя_пакета
```

для заведомо отсутствующего в системе пакета (опция -i указывает, что пакет должен быть инсталлирован, опция -h индицирует процесс распаковки пакета, а опция -v предписывает выводить текстовые сообщения о ходе инсталляции),

```
rpm -Fhv имя_пакета
```

для обновления явно имеющегося, и

```
rpm -Uhv имя_пакета
```

во всех сомнительных случаях, в том числе и при необходимости сохранения настроек.

Перед установкой пакетов, особенно взятых не из комплекта дистрибутива (а, например, скачанных из Интернета), лучше проверять их целостность командой

```
rpm -K file*.rpm
```

которая проверит контрольные суммы и цифровую подпись пакета и, при благоприятном результате, выдаст сообщение

```
file*.rpm: sha1 md5 gpg OK
```

К пакетам, не подписанным цифровой подписью gpg, следует относиться с особой осторожностью, так как в этом случае источник пакета проверить невозможно.

Запрос информации о пакете осуществляется в различных формах. Наиболее простая — rpm -q имя_пакета — выводит полное название последнего, включая все перечисленные ранее его компоненты. Например, ответом на

```
rpm -q ImageMagick
```

будет сообщение типа:

```
ImageMagick-5.2.7-1.asp
```

Дополнительные опции запроса делятся на опции выбора пакетов и опции выбора информации. К первым, помимо приведенного примера (где имя_пакета выступает не столько в качестве аргумента команды, сколько как ее опция), относятся:

- -a (--all) — запрос всех установленных пакетов;
- -f имя_файла (--file имя_файла) — запрос пакета, которому принадлежит файл имя_файла;
- -g имя_группы (--group имя_группы) — запрос пакетов из группы имя_группы;
- -p файл_пакета — запрос неустановленного пакета.

Опции выбора информации следующие:

- -i — вывод полной информации о пакете, включая название, версию и описание,
- -R (--requires) — вывод списка пакетов, от которых зависит данный пакет,
- -l (--list) — вывод списка файлов, входящих в данный пакет,
- -d (--docfiles) — вывод списка только файлов документации,
- -c (--configfiles) — вывод списка только конфигурационных файлов.

Для удаления пакета служит команда:

```
rpm -e имя_пакета
```

где -e — основная опция удаления, а в качестве аргумента достаточно просто названия, без номера версии. Вполне вероятно, что удаляемый пакет связан зависимостями с другими пакетами, установленными в системе. В этом случае приведенная команда не сработает, вызвав соответствующее сообщение. Если такой пакет все же необходимо удалить, следует прибегнуть к опции --nodeps — отказу от контроля зависимостей.

И для опций установки, и для опций удаления можно использовать дополнительную опцию `--test`. При ее указании соответствующие действия, определенные основной опцией, не выполняются, а только имитируются, в результате чего выводится сообщение о возможных нарушениях зависимостей, например:

```
rpm -e --test ImageMagick
```

ошибка: удаление этих пакетов нарушит зависимости:

```
ImageMagick нужен для xfig-3.2.3c-8
```

Компоненты пакетов, входящих в состав дистрибутива **ASPLinux**, обычно устанавливаются в подкаталоги каталога `/usr` (`/usr/bin`, `/usr/lib` и т.д.), права на запись в которые обычный пользователь не имеет. Кроме того, только суперпользователь имеет доступ на запись к базе данных установленных пакетов. Поэтому для использования `rpm` необходимы права суперпользователя. Тестовая установка или удаление могут быть выполнены и обычным пользователем.

С помощью `rpm` можно устанавливать и пакеты, не входящие в дистрибутив. В частности, бинарные пакеты, собранные для RedHat, почти во всех случаях будут успешно установлены под **ASPLinux**.

10.3 Установка исходных текстов программ из `rpm`-пакетов

Исходные тексты программ, включенных в состав дистрибутива **ASPLinux**, занимают два отдельных диска и часть на третьем установочном, где, как правило, располагаются в каталоге `SRPMS`. При просмотре их содержимого можно видеть файлы

```
ElectricFence-2.2.2-4.src.rpm
ImageMagick-5.2.2-5.src.rpm
Inti-0.5preview-1.src.rpm
```

и т.д., то есть rpm-пакеты, компонент `src` в имени которых указывает, что они включают не бинарные программы, а их исходные тексты. Для сборки таких пакетов также используется программа `rpm`, но с иными опциями — опциями сборки. Однако в первую очередь необходимо установить rpm-пакет с исходными текстами. Делается это точно так же, как и для бинарных rpm-пакетов, то есть с помощью команды

```
rpm -ihv имя
```

или программы **system-config-packages**. После этого в каталоге, отведенном под исходные тексты дистрибутива (/usr/src/asplinux/SOURCES), можно будет обнаружить новый архив имя.tar.gz и, скорее всего, несколько одноименных ему файлов вида имя.*.patch. Основной архив содержит собственно исходные тексты программы в том виде, в каком они распространяются ее разработчиком, а patch-файлы — дополнения и изменения, внесенные в них составителями дистрибутива для того, чтобы программа успешно компилировалась (и правильно работала) именно в данной системе.

Кроме того, в каталоге /usr/src/asplinux/SPECS появится файл вида имя.spec, содержащий данные о пакете в следующей форме:

```
Summary: An uncompressor for .arj format archive files.
Name: unarj
Version: 2.43
Release: 6
Group: Applications/Archiving
Copyright: distributable
Source: ftp://metalab.unc.edu/pub/linux/utils/compress/unarj%{version}.tar.gz
Patch: unarj-2.43-subdir.patch
BuildRoot: /var/tmp/unarj-root
```

его описание:

```
%description
The UNARJ program is used to uncompress .arj format archives.
The .arj format archive was mostly used on DOS machines.
Install the unarj package if you need to uncompress .arj format archives.
```

а главное порядок сборки бинарного пакета, накладывания патчей и т.д. Именно с этим файлом и будут производиться дальнейшие действия.

Основная опция сборки rpm-пакета -b, требующая минимум одной из дополнительных опций:

```
rpmbuild -bX имя.spec
```

где X соответствует одной из дополнительных опций. В числе их следующие:

- -p — выполнение стадии «%prep» spec-файла; обычно это распаковка исходных текстов и прикладывание к ним патчей;
- -l — выполнение «list check», то есть проверка файлов, перечисленных в секции «%files» spec-файла;

- `-c` — выполнение стадии «`%build`» spec-файла (с предварительным выполнением стадии «`%prep`»), то есть собственно компиляции пакета;
- `-i` — выполнение стадии «`%install`» spec-файла (с предварительным выполнением стадий «`%prep`» и «`%build`»), то есть размещение компонентов в соответствующие подкаталоги каталога `/usr/src/asplinux/`;
- `-b` — полная сборка бинарного файла с предварительным выполнением стадий «`%prep`», «`%build`» и «`%install`».

Из приведенного перечня можно видеть, что для сборки пакета в большинстве случаев следует воспользоваться командой `rpm` в следующей форме:

```
rpm -bb имя.spec
```

В результате в одном из подкаталогов каталога `/usr/src/asplinux/RPMS` (в зависимости от архитектуры, для которой пакет предназначен, скорее всего — в `/usr/src/asplinux/RPMS/i386`) появится файл вида `имя.*.rpm`, соответствующий бинарному пакету. Который может быть, наконец, установлен стандартным образом, то есть:

```
rpm -ihv имя.*.rpm
```

или

```
rpm -Uhv имя.*.rpm
```

и т.д., в зависимости от того, требуется установка его заново или лишь обновление. Существуют и другие способы установки rpm-пакетов с исходными текстами, с которыми можно ознакомиться посредством экранной документации:

```
man rpm
```

которая в дистрибутиве **ASPLinux** имеется в русскоязычном варианте.

10.4 Компиляция программ из исходных текстов

Сборка из исходных текстов программ, не входящих в дистрибутив, начинается с того, что архив с исходными текстами распаковывается в подходящий каталог (обычно для этого используются каталоги `/usr/local/src` или `$HOME/src`). После чего в нем появляется соответствующий имени программы подкаталог.

В правильно оформленной для распространения программе каталог этот должен содержать файлы README, INSTALL, Makefile, configure. Первые два содержат описание программы и процесса ее установки.

Файл Makefile описывает процесс сборки программы и указывает местонахождение необходимых для этого компонентов, в частности, системных библиотек. Обычно он ориентирован на некую усредненную конфигурацию, которая может не соответствовать (и, как правило, не соответствует) имеющимся реалиям.

Для приведения файла Makefile в соответствие с последними используется команда ./configure (по понятным причинам она обязательно запускается из текущего каталога).

После этого запускается программа make. Она производит сборку исходных текстов в т.н. объектные модули (нечто вроде оверлеев в DOS-программах). По завершении его, то есть возврату приглашения командной строки, собранную программу нужно установить, то есть записать исполнимые модули, библиотеки, документацию и прочее туда, где им надлежит быть впредь (как правило, по умолчанию это соответствующие подкаталоги /usr/local — /usr/local/bin, /usr/local/lib и т.д.). Для этого дается команда make install, которая и осуществляет этот процесс.

Наконец, завершающий шаг, необязательный, но крайне желательный — это освобождение каталога с исходными текстами от промежуточных файлов команды make, то есть объектных модулей. Делается это командой make clean.

В некоторых случаях для сборки программы достаточно одной команды, обычно make all или make install. В любом случае в первую очередь следует руководствоваться указаниями, данными в файлах документации (README или INSTALL).

10.5 Управление пакетами rpm при помощи Webmin

Управление rpm пакетами в Webmin осуществляет модуль «*Менеджер ПО*», расположенный в разделе «*Система*». При помощи этого модуля можно:

- Установить новые пакеты как из локальных файлов, так и из файлов, расположенных на ftp или http серверах.
- Осуществить поиск установленных пакетов.
- Определить, к какому пакету относится интересующий файл.

Установка пакета, находящегося на жестком диске, выполняется очень просто, необходимо указать имя файла и нажать на кнопку «**Установить**». Чтобы

ASPLinux Management Console:
Webmin
Система
Аутентификация PAM
Дисковые квоты
Файловые системы
Менеджер ПО
Монитор служб MON
Пароли
Пользователи и группы
Процесс инициализации (init)
Процессы
Расписание заданий
Расписание заданий Cron
Резервное копирование
Система безопасности
Системный журнал
Страницы руководства
Загрузка и завершение работ
Службы
Сеть
Оборудование
Прочее
Change Password
Выйти

Справка. Пакеты программного обеспечения Поиск в справке.

Установленные пакеты

Искать пакет Дерево пакетов

Установить новый пакет

Выберите, откуда устанавливать новый пакет.: RPM

Из локального файла

Из загруженного файла Browse...

С ftp или http URL

Установить

Определить принадлежность файла

Укажите команду или путь к файлу, который вы хотите найти в базе данных RPM.

Искать

Рис. 10.1: Управление пакетами rpm при помощи Webmin

установить rpm пакет, расположенный на CD-ROM, следует вставить CD-ROM в привод. Затем в модуле «*Файловые системы*», находящемся в разделе «*Система*», убедитесь, что CD-ROM подмонтирован — в столбце «*Подмонтирована?*» в строке /mnt/cdrom должно быть написано Да. Если он не подмонтирован, тогда в столбце будет написано Нет, в этом случае нажмите на «**Нет**» и CD-ROM будет подключен. Перейдите в модуль «*Менеджер ПО*» и в разделе «*Установка Нового Пакета*» выберите «*Локального файла*». Для выбора файла пакета, следует нажать на кнопку «**...**». В появившемся окне необходимо дважды кликнуть на директории mnt, затем на директории cdrom и в нем будет показано содержимое CD-ROM. Выберите необходимый пакет (файл с расширением rpm) и нажмите «**OK**». В поле ввода будет показан полный путь к файлу.

Для установки пакета следует нажать на кнопку «**Установить**». После установки следует извлечь CD-ROM из привода, но сначала его необходимо отмонтировать в модуле «*Файловые системы*».

Для поиска пакета необходимо ввести его имя в поле ввода и нажать кнопку «**Искать пакет**». Можно использовать не полное имя пакета, а часть слова, например, python. В результате поиска будет показан список пакетов, в названии которых присутствует слово python.

Кнопка «**Дерево пакетов**», расположенная на главной странице модуля, показывает список всех пакетов, выполненный в виде «дерева». При нажатии на пиктограмму «папка» будет показан список пакетов — содержимое «ветки» дерева. При повторном нажатии список будет спрятан.

Если в списке пакетов выбрать ссылку с именем пакета, будет показана страница с информацией о нем. При нажатии на кнопку «**Просмотр файлов**», на

новой странице будет сформирован список файлов, принадлежащих пакету. А если нажать на кнопку «**Удалить**» — пакет будет удален.

В разделе «*Определить принадлежность файла*», главной страницы модуля, в поле ввода можно в вести имя файла и нажать на кнопку «**Искать**». В результате поиска, будет показан пакет, в состав которого входит искомый файл. Для поиска необходимо указывать полный путь к файлу. Кнопка «**...**» облегчает ввод имени файла. В появившемся окне выберите интересующий файл и нажмите на кнопку «**OK**».

10.6 Автоматическое обновление системы при помощи Yum

В состав дистрибутива **ASPLinux** входит **Yum** — система автоматической установки, удаления и обновления пакетов **RPM**. Она автоматически учитывает имеющиеся или возникающие в процессе установки или обновления пакетов зависимости и делает этот процесс прозрачным для пользователя.

Yum копирует заголовки из пакетов **RPM** с сервера (называемого репозиторием и представляющего собой HTTP-, ftp- сервер или, в случае если используется сборка **Yum**, поддерживающая протокол `file://`, просто место на диске) в свою рабочую область на клиентской машине. Когда требуется провести какое-то действие, то все процессы разрешения/выявления зависимостей и т.п. производятся непосредственно на клиентской машине, определяя, что же необходимо установить/удалить/обновить.

Yum поддерживает работу через прокси-сервер. Для использования прокси Вам необходимо настроить переменную окружения `http_proxy` на прокси-сервер, например так:

```
http_proxy="http://www.someproxy.com:3128"
export http_proxy
```

После чего **Yum** сможет использовать прокси-сервер.

С первым запуском **Yum** буду загружены заголовки с сервера (локального или удаленного), которые впоследствие будут обновляться лишь при обновлении содержимого репозитория.

10.6.1 Основные команды при работе с Yum

- Просмотреть список всех доступных к установке пакетов:

  `yum list`

- Этой команды достаточно для просмотра, иногда лучше воспользоваться командой:

```
yum list|less
```

- Просмотреть только список обновлений:

```
yum list updates
```

- Список установленных пакетов:

```
yum list installed
```

В каждой из этих команд можно использовать дополнительный параметр: имя или шаблон имени пакета. Например, yum list kernel*[1] покажет список доступных пакетов начинающихся с 'kernel'.

Списки имеют унифицированный формат.

```
[root@andriy ~]# yum list kernel\*
Setting up Repos
base                     100% |=========================|  903 B 00:00
updates                  100% |=========================|  951 B 00:00
Reading repository metadata in from local files
base      : ################################################## 2146/2146
updates   : ################################################## 1168/1168
Installed Packages
kernel.i686                              2.6.11-1.35asp installed
kernel-utils.i386                        1:2.4-13.1.49_FC3 installed
Available Packages
kernel.i586                              2.6.11-1.35asp updates
kernel-doc.noarch                        2.6.11-1.35asp updates
kernel-smp.i686                          2.6.11-1.35asp updates
kernel-smp.i586                          2.6.11-1.35asp updates
```

Зачастую бывает нужно установить пакет не зная его имени, а имея в распоряжении только имя какого-нибудь файла, который должен принадлежать этому пакету. Допустим, нам нужно найти, в каком пакете находится нужная нам библиотека, выполнив команду yum provides /usr/lib/libcurl.so:

```
...

curl.i386                                7.12.1-1 base
Matched from:
/usr/lib/libcurl.so.3.0.0
/usr/lib/libcurl.so.3

curl-devel.i386                          7.12.1-1 base
Matched from:
/usr/lib/libcurl.so
```

[1] Если в текущем каталоге приступвуют файлы kernel*., то символ ≪*≫ надо заэкранировать, указав т.о. последовательность ≪*≫

```
curl.i386                                7.12.3-3.fc3 updates
Matched from:
/usr/lib/libcurl.so.3
/usr/lib/libcurl.so.3.0.0


curl-devel.i386                          7.12.3-3.fc3 updates
Matched from:
/usr/lib/libcurl.so


curl.i386                                7.12.3-3.fc3 installed
Matched from:
/usr/lib/libcurl.so.3
/usr/lib/libcurl.so.3.0.0


curl-devel.i386                          7.12.3-3.fc3 installed
Matched from:
/usr/lib/libcurl.so
```

Из чего видно, что пакет, содержащий такой файл, уже уставлен и называется curl-devel.

Не всегда есть возможность скачать и установить все что надо, поэтому прежде чем устанавливать новые пакеты, всегда можно прочитать информацию о них командой `yum info mozilla`:

```
...
Installed Packages
Name   : mozilla
Arch   : i386
Version: 1.7.8
Release: 1.3.1asp
Size   : 33 M
Repo   : installed
Summary: Браузер Web.

Description:
 Mozilla - это браузер www с открытым исходным кодом, созданный по
высоким стандартам производительности, скорости работы и возможностью
портирования.
```

10.6.2 Удаление, обновление и установка пакетов с помощью Yum

Существует три режима установки и обновления пакетов при помощи **Yum**

- Удаление пакетов
- Установка пакетов
- Обновление пакетов

Например, нужно удалить пакет php (командой yum remove php):

```
Setting up Remove Process
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package php.i386 0:4.3.11-2.6 set to be erased
--> Running transaction check
Setting up Repos
base                    100% |=========================|  903 B
00:00
updates                 100% |=========================|  951 B
00:00
Reading repository metadata in from local files
base      : ################################################## 2146/2146
updates   : ################################################## 1168/1168
--> Processing Dependency: php = 4.3.11-2.6 for package: php-pear
--> Processing Dependency: php = 4.3.11-2.6 for package: php-devel
--> Restarting Dependency Resolution with new changes.
--> Populating transaction set with selected packages. Please wait.
---> Package php-devel.i386 0:4.3.11-2.6 set to be erased
---> Package php-pear.i386 0:4.3.11-2.6 set to be erased
--> Running transaction check

Dependencies Resolved
Transaction Listing:
  Remove: php.i386 0:4.3.11-2.6

Performing the following to resolve dependencies:
  Remove: php-devel.i386 0:4.3.11-2.6
  Remove: php-pear.i386 0:4.3.11-2.6
Total download size: 0
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Erasing: php 1/3
Erasing: php-devel 2/3
Erasing: php-pear 3/3

Removed: php.i386 0:4.3.11-2.6
Dependency Removed: php-devel.i386 0:4.3.11-2.6 php-pear.i386 0:4.3.11-2.
Complete!
```

Yum выяснил, что для сохранения зависимостей нужно удалить еще и все зависимые от php пакеты.

Теперь php удален.

Установка пакетов — самый простой процесс. Пример команды yum install php следует ниже:

```
...

Parsing package install arguments
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package php.i386 0:4.3.11-2.6 set to be updated
--> Running transaction check
--> Processing Dependency: php-pear for package: php
--> Restarting Dependency Resolution with new changes.
--> Populating transaction set with selected packages. Please wait.
---> Package php-pear.i386 0:4.3.11-2.6 set to be updated
--> Running transaction check

Dependencies Resolved
Transaction Listing:
  Install: php.i386 0:4.3.11-2.6 - updates

Performing the following to resolve dependencies:
  Install: php-pear.i386 0:4.3.11-2.6 - updates
Total download size: 1.6 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Installing: php-pear 100 % done 1/2
Installing: php 100 % done 2/2

Installed: php.i386 0:4.3.11-2.6
Dependency Installed: php-pear.i386 0:4.3.11-2.6
Complete!
```

Совершенно необязательно указывать весь список нужных пакетов. Например если указать только php-pgsql, то **Yum** сам определит, что необходимо установить также и php.

Для обновления пакетов используется команда yum update. Например, для обновления пакета curl необходимо выполнить команду yum update curl:

```
...

Resolving Dependencies
```

```
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for curl to pack into transaction set.
curl-7.12.3-3.fc3.i386.rp 100% |=========================| 9.8 kB 00:00
---> Package curl.i386 0:7.12.3-3.fc3 set to be updated
--> Running transaction check
--> Processing Dependency: curl = 7.12.1-1 for package: curl-devel
--> Restarting Dependency Resolution with new changes.
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for curl-devel to pack into transaction set.
curl-devel-7.12.3-3.fc3.i 100% |=========================|  19 kB 00:00
---> Package curl-devel.i386 0:7.12.3-3.fc3 set to be updated
--> Running transaction check

Dependencies Resolved
Transaction Listing:
  Update: curl.i386 0:7.12.3-3.fc3 - updates

Performing the following to resolve dependencies:
  Update: curl-devel.i386 0:7.12.3-3.fc3 - updates
Total download size: 503 k
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Updating: curl 100 % done 1/4
Updating: curl-devel 100 % done 2/4
Completing update for curl-devel  - 3/4
Completing update for curl  - 4/4

Updated: curl.i386 0:7.12.3-3.fc3
Dependency Updated: curl-devel.i386 0:7.12.3-3.fc3
Complete!
```

Для обновления всего дистрибутива нужно воспользоваться командой yum update без указания последующих параметров.

10.6.3 Настройка репозитория

Разложите ваши **RPM**-пакеты внутри одного каталога. Ничего страшного, если в нем будут подкаталоги, поскольку обычно администраторы стараются держать i386 и noarch пакеты в разных подкаталогах.

Чтобы построить репозиторий, воспользуйтесь командой[2]

createrepo -z имя_каталога_репозитория

Добавьте запись о репозитории в файле с расширением .repo, созданном

[2]Она располагается в одноименном пакете createrepo.

в каталоге /etc/yum.repos.d на клиентской машине. Например, если вы создали репозиторий в каталоге /var/my-repository, то в файл настроек, который будет наподобие /etc/yum.repos.d/mylocal.repo, надо вписать следующие строки:

```
[mylocal]
name=Мой собственный репозиторий
baseurl=file:///var/my-repository
```

Глава 11

Сборка ядра системы

Для дистрибутивов Linux давностью несколько лет перекомпиляция ядра сразу после установки практически была необходима: их ядра, по умолчанию рассчитанные на поддержку некоей конфигурации, с одной стороны, отягощали машины, не отличавшиеся еще избыточной мощностью, с другой — не поддерживали многие необходимые устройства.

Ныне ситуация иная. Большинство современных дистрибутивов, и **ASPLinux** — яркий тому пример, по умолчанию устанавливаются с ядрами, обеспечивающими работу практически всех распространенных устройств общего назначения — звуковых плат, и записывающих приводов CD-R/RW, часто даже плат видеозахвата и TV тюнеров. В результате пользователь сразу после установки получает в свое распоряжение систему, поддерживающую и использующую если не все, то по меньшей мере большинство устройств, содержащихся в компьютере.

Поэтому разработчиками **ASPLinux** перекомпиляция ядра в принципе не рекомендуется без веских на то оснований, особенно для начинающих пользователей. Прибегать к ней следует только в крайних случаях, и только при понимании того, зачем это делается: ради поддержки новых устройств и функций, нестабильности работы в существующей конфигурации, и т.д. Поэтому данная процедура описывается в настоящем руководстве.

11.1 Версия и пакет ядра

ASPLinux основан на ядре Linux версии 2.6.x. Это ядро включает множество дополнительных заплат для исправления ошибок и добавления дополнительных функций. По этому ядро **ASPLinux** не может быть полным эквивалентом так называемого vanilla kernel с сайта http://www.kernel.org. Полный список этих заплат можно получить следующей командой:

```
rpm -qpl kernel-<version>.src.rpm
```

11.2 Варианты сборки ядра

ASPLinux поставляется с несколькими вариантами сборки ядра, такими как: обычной, smp и hugemem. Обычная сборка предназначена для однопроцессорных машин и поддерживает до 4ГБ ОЗУ. Сборка smp предназначена для многопроцессорных машин и поддерживает до 16ГБ ОЗУ. Сборка hugemem предназначена для машин с очень большим объемом ОЗУ и поддерживает до 64ГБ ОЗУ.

Для сборки некоторых модулей могут понадобиться отдельные файлы из исходных текстов ядра. Эти файлы содержатся в пакетах kernel-devel-<version>.<arch>.rpm.

Одновременно можно установить пакеты kernel-devel для нескольких вариантов сборки ядра. При этом они будут установлены в каталоге /usr/src/kernels/<version>-<arch>.

Во многих учебниках и примерах подразумевается, что исходные тексты ядра должны быть установлены в каталоге /usr/src/linux. Если вы создадите следующую символическую ссылку, вы сможете использовать эти материалы с **ASPLinux**.

```
ln -s /usr/src/kernels/kernel-<all-the-rest> /usr/src/linux
```

11.3 Подготовка к пересборке ядра

В отличие от предыдущих версий, **ASPLinux** не включает исходные тексты ядра. Пользователи, желающие получить доступ к оригинальным исходным текстам ядра из **ASPLinux**, могут найти их в соответствующей версии пакета kernel-<version>.src.rpm, который находится на одном из дисков с исходными текстами. Для получения развернутого дерева исходных текстов ядра нужно выполнить следующую последовательность команд.

Найти пакет kernel-<version>.src.rpm, соответствующий текущему ядру, на одном из дисков с исходными текстами или в каталоге SRPMS на сайте обновлений **ASPLinux**. Версия текущего ядра определяется командой uname -r.

Установить пакет kernel-<version>.src.rpm командой

```
rpm -Uvh kernel-<version>.src.rpm
```

Подготовить исходные тексты к сборке следующими командами:

```
cd /usr/src/asplinux/SPECS
rpmbuild -bp --target $(arch) kernel.spec
```

Дерево исходных текстов ядра будет развернуто в каталоге /usr/src/asplinux/BUILD/kernel-<version>.

Для соответствия общедоступной документации это дерево исходных текстов можно переместить в /usr/src при помощи следующей последовательности команд:

```
cd /usr/src/asplinux/BUILD/kernel-<version> /usr/src/
mv linux-<version> /usr/src/
cd /usr/src
ln -s ./linux-<version> linux
cd /usr/src/linux
```

Файлы конфигурации для отдельных ядер, поставляемых с **ASPLinux**, находятся в каталоге configs. Например, имя файла конфигурации для варианта сборки i686 SMP будет configs/kernel-<version>-i686-smp.config.

Поместите его в необходимое для пересборки место следующей командой:

```
cp configs/<desired-config-file> .config
```

Введите следующую команду:

```
make oldconfig
```

После этого можно продолжить стандартную процедуру сборки ядра.

11.4 Подготовка к конфигурированию ядра

Под конфигурированием ядра понимается включение в него или, соответственно, отключение поддержки всякого рода устройств и файловых систем. Кроме того, ряд опций может быть сконфигурирован как модули. То есть непосредственно в ядро они не встраиваются (для уменьшения его размера), но подгружаются по мере необходимости, автоматически или специальной программой

```
modprobe имя_модуля
```

Для конфигурирования ядра в Linux предусмотрены специальные утилиты. Они основаны на стандартной программе make и позволяют настроить опции ядра перед компиляцией. Это команды:

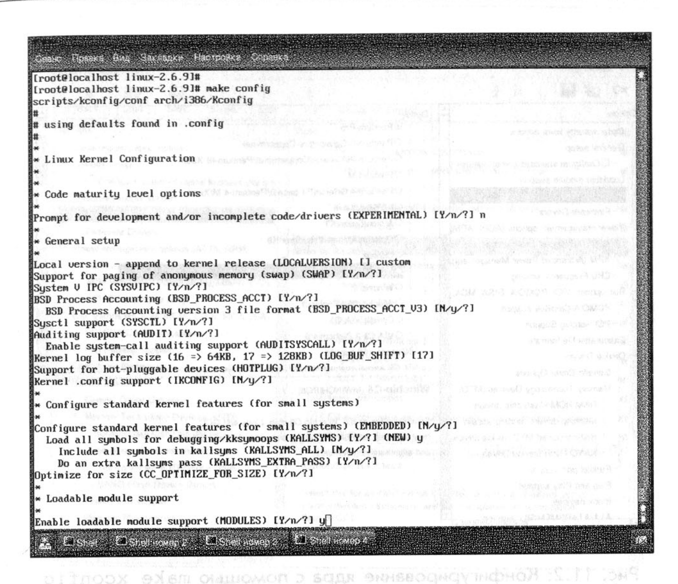


Рис. 11.1: Конфигурирование ядра с помощью `make config`

```
make config
make menuconfig
make xconfig
```

Все они выполняются от лица суперпользователя в каталоге, корневом для исходных текстов данной версии ядра, т.е. каталоге `/usr/src/linux-<version>`.

11.5 Средства конфигурирования ядра

Перейдем к рассмотрению средств конфигурирования. Первым из них идет `make config`. Это текстовая утилита, задающая в интерактивном режиме множество вопросов, на которые даются ответы: `yes` для включения опции в ядро, `no` — для ее исключения и `m` — для оформления в виде подгружаемого модуля (рис. 11.1).

Вариант ответа, первый символ которого дан в верхнем регистре, представляет собой умолчание. Чтобы согласиться с ним, достаточно нажать `Enter`. Набрав вместо ответа вопросительный знак, можно получить комментарий по поводу данной опции.

Главный недостаток этого способа конфигурирования — невозможность вне-

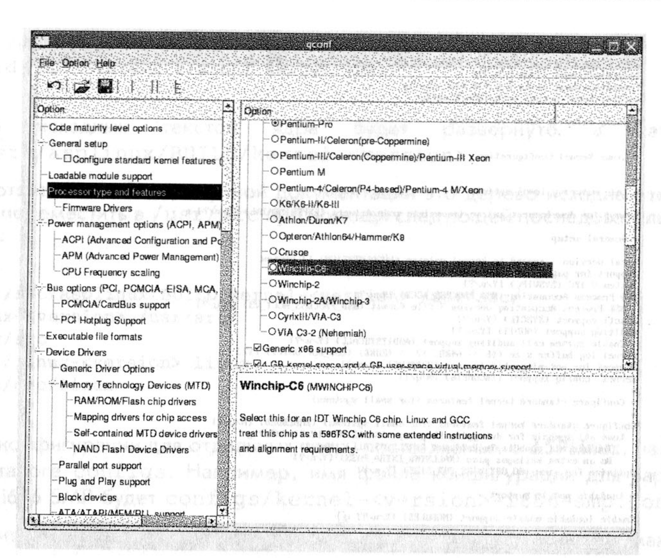


Рис. 11.2: Конфигурирование ядра с помощью make xconfig

сти изменения в течение текущей сессии. Любая ошибка при ответах на вопросы требует останова программы (с помощью, например, Ctrl+C) и начала процесса заново. Введенные ранее ответы при этом не сохраняются — все опять идет от конфигурации по умолчанию.

Другой недостаток make config — сложность использования готовых конфигурационных файлов (вроде упомянутых выше) как основы.

На другом полюсе — использование make xconfig. Это, напротив, программа графического режима, запускаемая из X Window System из окна эмуляции терминала. Следует напомнить, однако, что перед ее запуском нужно в терминальном же режиме перейти в каталог с исходными текстами ядра — иначе последует сообщение об ошибке.

Здесь группы вопросов о конфигурации оформлены в виде кнопок, нажатие на которые вызывает панели с дополнительными вопросами (рис. 11.2).

Преимущество make xconfig перед make config — в возможности считать некий предварительный файл конфигурации и возможности возврата к любому из пройденных шагов, существенный недостаток — необходимость загрузки X Window System, что не всегда желательно. Поэтому оптимальный способ конфигурирования ядра — с помощью make menuconfig. Эта утилита работает в текстовом режиме, но имеет псевдографический, интерактивно управляемый интерфейс, дающий возможность вернуться к уже пройденным группам вопросов и внести необходимые коррективы в случае ошибки.

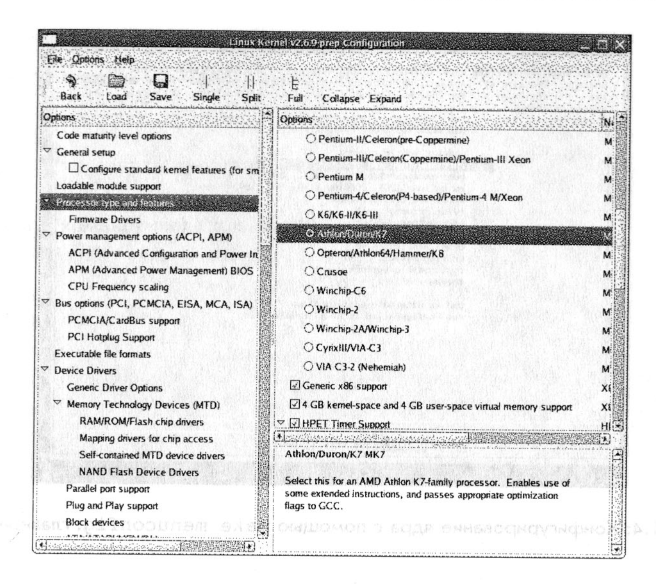


Рис. 11.3: Использование make gconfig

По умолчанию для make xconfig используется библиотека Qt. Если в системе таковая отсутствует, то рекомендуется в таком случае воспользоваться опцией gconfig, которая вызовет оболочку, использующую библиотеку Gtk. Ее стандартный вид представлен на рисунке рис. 11.3.

Начало работы — переход в каталог /usr/src/linux-<version>, и набор в командной строке (в режиме администратора) команды

```
make menuconfig
```

Возникает главное меню конфигурации ядра, с рядом пунктов (рис. 11.4), которые рассмотрены подробно в специальной литературе.

11.6 Стратегия конфигурирования

Рассмотрев опции конфигурации ядра, целесообразно вернуться к вопросам ее стратегии. Обычно дается две рекомендации по этому поводу. Первая — во всех сомнительных или неясных случаях избирать вариант, предложенный по умолчанию. Вторая — включать поддержку не только тех устройств, которые есть, но и тех, которые, возможно, будут установлены в будущем, чтобы после этого не заниматься переконфигурированием ядра заново.

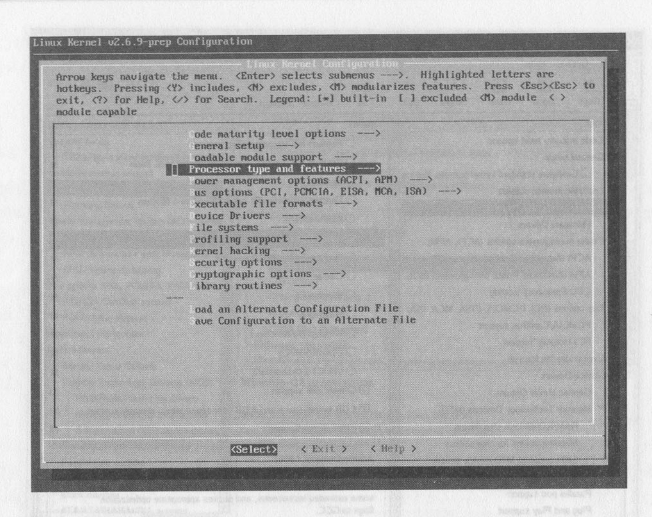


Рис. 11.4: Конфигурирование ядра с помощью `make menuconfig`: главное меню

Обе рекомендации оправданы для компьютера, используемого в качестве сервера. Однако для настольного применения целесообразным представляется исключение поддержки всех отсутствующих устройств и всех опций, необходимость которых вызывает сомнение.

Первое, что может грозить в этом случае — ядро откажется компилироваться, или при сборке модулей последует сообщение об ошибке. Однако внимательное чтение вывода на экран обычно позволяет локализовать причину сбоя, после чего можно вернуться к конфигурированию и внести соответствующие коррективы.

Вторая потенциальная опасность — успешно, казалось бы, скомпилированное ядро откажется запускаться. Для предотвращения этого достаточно скопировать предыдущее, работоспособное, ядро под другим именем и внести в начальный загрузчик (LiLo или ASPLoader) изменения, позволяющие выбрать его загрузку как альтернативу.

11.7 Сборка только модулей ядра

Развернутое дерево исходных текстов ядра больше не требуется для сборки модулей ядра, таких как ваш собственный драйвер устройства.

Например, для сборки модуля `foo.ko` нужно создать следующий `Makefile.in` в каталоге, содержащем файл `foo.c`:

```
obj-m := foo.o

KDIR  := /lib/modules/$(shell uname -r)/build
PWD   := $(shell pwd)

default:
	$(MAKE) -C $(KDIR) M=$(PWD) modules
```

Используйте команду make для сборки модуля foo.ko.

Глава 12

Администрирование сети

В задачи администрирования сети входят:

- настройка сетевых протоколов и сетевых интерфейсов;
- настройка системы доменных имен и сетевой маршрутизации;
- настройка различных сетевых сервисов (почтовых, web, proxy-сервера и т.д.).

Все эти вопросы будут рассмотрены в настоящей главе. Однако предварительно следует остановиться на общих представлениях о базовом сетевом протоколе Linux — протоколе IP, и об основных сетевых интерфейсах, используемых в этой операционной системе.

12.1 Общие сведения об Internet Protocol

`Internet Protocol` (IP) был создан в 70-х годах для поддержки ранних локальных сетей с ОС Unix. В настоящий момент IP пришел как стандарт для всех современных сетевых ОС для коммуникации друг с другом. Многие популярные высокоуровневые протоколы типа HTTP и TCP базируются на IP.

В операционной системе Linux базовым сетевым протоколом является протокол IP. Поддерживаются и другие сетевые протоколы, но они не являются базовыми, и потому не рассматриваются в настоящем руководстве.

Передача данных посредством протокола IP производится в пакетном режиме. При этом каждый пакет, передаваемый по коммуникационным каналам, содержит адрес отправителя и адрес получателя.

12.2 Система IP адресов

В промышленном использовании на сегодняшний день находятся две версии IP. Ранее, все сети использовали IP версии 4 (IPv4), но в связи с увеличением числа учебных и исследовательских сетей они стали адаптироваться в следующее поколение IP версии 6 (IPv6).

12.2.1 Адресная нотация IPv4

Адрес IPv4 состоит из четерех байт (32 бита). Эти байты также известны как октеты.

Для большей читаемости, типовое использование IP адресов предлагается в виде десятичной нотации, где каждый октет разделен символом . (точка). Например, IP адрес

```
00001010 00000000 00000000 00000001
```

обычно представляется в эквивалентной десятичной записи 10.0.0.1

Поскольку каждый байт длиной в 8 бит, каждый октет в IP адресе представим значением в диапазоне от 0 до 255. Следовательно, полный диапазон IP адресов от 0.0.0.0 влоть до 255.255.255.255, что представляет в сумме 4 294 967 296 возможных IP адресов.

12.2.2 Адресная нотация IPv6

IP адресация сильно изменилась с появлением IPv6. Адреса IPv6 уже состоят из 16 байт (128 бит), что намного длиннее четырех байт (32 бита). А это уже содержит более чем 300e+48 возможных адресов! В ближайшие годы при стабильном увеличении числа сотовых телефонов, КПК и других устройств, работающих с сетями, будет расширятся их сетевая емкость, а это вероятно потребует большего адресного пространства IPv6.

Адреса IPv6 в основном записываются в следующей форме:

```
hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh
```

В этой нотации пары байт IPv6 разделяются двоеточием и каждый байт раскрывается парой шестнадцатиричных цифр, как показано ниже:

```
E3D7:0000:0000:0000:51F4:9BC8:C0A8:6420
```

Адреса IPv6 часто содержат множество байт из нулевых значений. Укороченная нотация в IPv6 удаляет эти значение из текстового представления (хотя байты все еще присутствуют в актуальном сетевом адресе) как показано далее:

Класс	Левые 4 бита	Адреса	
A	0xxx	0.0.0.0	127.255.255.255
B	10xx	128.0.0.0	191.255.255.255
C	110x	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E	1111	240.0.0.0	255.255.255.255

Таблица 12.1: Классы IP адресов

```
E3D7::51F4:9BC8:C0A8:6420
```

Наконец, многие IPv6 адреса являются расширением пространства адресов IPv4. В этих случаях самая правая часть из четырех байт в адресе IPv6 (самые правые две пары байт) может быть переписана в нотации IPv4. Переведенный выше пример в смешанной нотации дает

```
E3D7::51F4:9BC8:192.168.100.32
```

12.2.3 Классы адресов IPv4

Адресное пространство IPv4 может быть поделено на 5 классов - A, B, C, D и E. Каждый класс состоит из непрерывного подмножества внутри всего диапазона адресов IPv4.

С несколькими специальными исключениями, разъясненными ниже, значения левых четырех бит адреса IPv4 определяют его класс как показано в таблице 12.1.

Все адреса класса C, к примеру, имеют левые три бита, установленные в '110', но каждый из оставшихся 29 бит может быть установлен либо в '0', либо в '1' независимо (как представлено символом 'x' в этих битовых полях):

```
110xxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

Переведя вышеуказанное в точечно-десятичную нотацию, это даст, что все адреса класса C располагаются в диапазоне от 192.0.0.0 вплоть до 223.255.255.255.

12.2.4 Класс IP адресов E и ограниченное широковещание

Сетевой стандарт IPv4 определяет адреса класса E как зарезервированные, подразумевая, что они не должны использоваться в IP сетях. Некоторые исследовательские организации используют адреса класса E для экспериментальных целей. Однако, узлы, пытающиеся использовать эти адреса в Интернете, не смогут корректно обмениваться данными.

Специальный тип IP адреса — это т.н. ограниченный широковещательный адрес 255.255.255.255. Широковещательный запрос выполняет доставку сообшения от отправителя ко многим получателям. Отправители направляют запрос прямо к 255.255.255.255 чтобы показать, что все остальные узлы в локальной сети (LAN) должны принять сообщение. Этот широковещательный запрос будет 'ограниченным' в том, что он достигнет лишь узлов LAN, а не всех узлов в Интернете.

Технически IP резервирует целиком диапазон адресов от 255.0.0.0 вплоть до 255.255.255.255 для широковещания, и это диапазон не должен рассматриваться ка часть нормального диапазона класса E.

12.2.5 Класс IP адресов D и многоадресное вещание

Сетевой стандарт IPv4 определяет класс D адресов как зарезервированный для многоадресного вещания. Многоадресное вещание — это механизм для определения групп узлов и отправки IP сообщений в эти группы, вместо отправки каждому узлу в LAN (широковещание) или только для одного адресата (одноадресное вещание).

Многоадресное вещание в основном использовалось в исследовательских сетях. Как с адресами класса E, класс D не должен использоваться обычными узлами в Интернете.

12.2.6 IP адреса классов A, B и класса C

Классы A, B и класс C — это три класса адресов, используемых в IP сетях общего назначения вместе с тремя исключениями, описанными далее.

12.2.7 IP адрес кольцевого интерфейса

127.0.0.1 — это адрес кольцевого интерфейса в IP. Кольцевой интерфейс представляет собой тестовый механизм сетевых адаптеров. Сообщения, посланные 127.0.0.1, не доставляются в сеть. Вместо этого адаптер перехватывает все сообщения кольцевого интерфейса и возвращает их пославшему приложению. Приложения IP часто используют эту особенность для проверки поведения их сетевого интерфейса.

Как и с широковещательным адресом, IP официально резервирует весь диапазон с 127.0.0.0 по 127.255.255.255 для кольцевого интерфейса. Узлы не должны использовать этот диапазон в Интернете и это не должно рассматриваться как часть нормального диапазона адресов класса A.

Класс	Количество сетей	Маска сети	Адреса	
A	1	255.0.0.0	10.0.0.0	10.255.255.255
B	16	255.240.0.0	172.16.0.0	172.31.255.255
C	256	255.255.0.0	192.168.0.0	192.168.255.255

Таблица 12.2: IP адреса для частного использования

12.2.8 Нулевые адреса

Как и с диапазоном кольцевого интерфейса диапазон адресов от 0.0.0.0 вплоть до 0.255.255.255 не должен рассматриваться как часть нормального диапазона класса A. Адреса вида 0.x.x.x не выполняют какую-то особенную роль в IP, но узлы, пытающиеся использовать их, не получат корректного обмена в Интернет.

12.2.9 Частные адреса

IP стандарт[1] определяет специальные диапазоны адресов внутри классов A, B и класса C, зарезервированных для использования в частных сетях (интранет). Таблица 12.2, приведенная ниже, описывает эти зарезирвированные диападоны адресного пространства IP.

Этими IP адресами можно беспрепятственно пользоваться для построения локальных, корпоративных и образовательных сетей, неимеющих выхода в Internet, или расположенных за брандмаурами, или другими шлюзами, использующими перевод сетевых адресов (от англ. `Network Address Translation или NAT`). Кроме того, именно они используются для организации сетевого взаимодействия на локальном компьютере между виртуальными машинами **VMWare**.

12.2.10 Типы адресов IPv6

IPv6 не использует классы. IPv6 поддерживает следующие три типа IP адресов:

- unicast — одноадресное вещание;
- multicast — многоадресное вещание;
- anycast — вещание на любой адрес.

Одноадресная и многоадресная передача сообщений в IPv6 концептуально точно такая же, как и в IPv4. IPv6 не поддерживает широковещание, так как

[1] См. также rfc1819.

его механизм многоадресной доставки соответствует в высокой степени тому же эффекту. Многоадресная нотация в IPv6 начинается с 'FF' (255) как и в IPv4 адресах.

Вещание на любой адрес в IPv6 является вариацией многоадресной доставки. Тогда как многоадресное вещание выполняет доставку сообщений ко всем узлам многоадресной группы, вещание на любой адрес доставляет сообщения к одному любому адресату в многоадресной группе. Вещание на любой адрес — это проект продвинутой сетевой концепции для поддержки преодоления отказов и баланса загрузки, необходимых приложениям.

12.2.11 Зарезервированные адреса IPv6

IPv6 резервирует только два специальных адреса: 0:0:0:0:0:0:0:0 и 0:0:0:0:0:0:0:1. IPv6 использует 0:0:0:0:0:0:0:0 для внутренних целей в реализации протокола, так узлы не могут использовать его для своих собственных коммуникационных целей. IPv6 использует 0:0:0:0:0:0:0:1 как адрес кольцевого интерфейса, эквивалентный к 127.0.0.1 в IPv4.

12.2.12 Сетевое разделение IP

Компьютерные сети состоят из индивидуальных сегментов сетевого кабеля. Электрические свойства кабеля ограничивают полезный объем любого заданного сегмента, так что даже скромные локальные сети требуют нескольких сегментов. Шлюзовые устройства типа маршрутизаторов и мостов соединяют эти сегменты вместе, хотя и не совершенно однородно.

За пределами разделения из-за использвоания кабеля лежит подразделение сети на подсети. Подсети поддерживают виртуальные сетевые сегменты, которые дробят трафик, протекающий по кабелю, на несколько частей. Конфигурация подсети часто совпадает с расположением сегмента один-в-один, но подсети могут быть также поделены на заданные сетевые сегменты.

12.2.13 Нумерация IP сети

Даже без деления на подсети (разъяснено ранее), хост-узлы в Internet или других other IP сетях получают сетевой номер. Сетевой номер позволяет группе хостов (потребители) обмениваться данными эффективно друг с другом. Хосты в той же сети могут быть компьютерами, размещенными в том же самом здании, или компьютерами, используемыми рабочей группой, например. Хосты, имеющие несколько сетевых адаптеров, (т.н. multi-homed) могут оперировать с несколькими сетями, но каждый адаптер привязан точно к одному сетевому номеру.

Класс	Сетевой адрес	Маска по умолчанию	Адреса хостов	
A	x.0.0.0	255.0.0.0	0.0.0.0	127.255.255.255
B	x.x.0.0	255.255.0.0	128.0.0.0	191.255.255.255
C	x.x.x.0	255.255.255.0	192.0.0.0	223.255.255.255

Таблица 12.3: Сетевая нумерация IP

Номера сетей выглядят очень похоже на IP адреса, но их следует отличать. Рассматриваемый пример с хостом, имеющим IP адрес 10.0.0.1, используется в основном в частных сетях. Поскольку этот адрес из класса A без разбиения на подсети, его самый левый байт (восемь бит) по умолчанию ссылаются на сетевой адрес, а все остальные биты установлены в ноль. Отсюда, 10.0.0.0 является сетевым номером, соответствующим IP адресу 10.0.0.1.

Часть IP адреса, что не ссылается на сеть, вместо этого отвечает за адрес хоста — буквально это уникальный идентификатор хоста в данной сети. В приведенном выше примере адрес хоста выглядит как '0.0.0.1' или просто '1'. Также необходимо заметить, что адрес сети выглядит как зарезервированный адрес, который не должен быть присвоен ни одному реальному хосту. Настройка живого хоста на 10.0.0.0 в примере, приведенном выше, ударит по коммуникациям для всех хостов той сети.

Ниже приведена таблица 12.3, иллюстрирующая схему нумерации по умолчанию для сетей классов A, B и C.

Главное, сетевые адреса задействуют левый байт для адресации их хостов, если хосты попадают внутрь диапазона класса A, левые два байта для хостов в классе B и левые три байта для хостов их класса C. Такой алгоритм прилагается на практике при оперировании с сетевыми масками. Вышеприведенная таблица показывает десятичное представление сетевых масок по умолчанию, которые в основном используются сетевыми ОС. Следует отметить, что десятичное значение '255' ссылается на один байт, который имеет установленные в единицу все биты (11111111).

12.2.14 Выгода от сетевой адресации

Сетевая адресация фундаментально организовывает хосты в группы. Это способно улучшить безопасность (путем изолирования критических ущлов) и может снизить сетевой трафик (путем предотвращения передачи между узлами, которые не должны взаимодействовать друг с другом). Помимо всего, сетевая адресация получается даже более мощной, когда представляется подсетевым делением и надсетевым.

12.2.15 CIDR - безклассовая междоменная маршрутизация

CIDR — это аббревиатура от английской фразы `Classless Inter-Domain Routing`. CIDR была разработана в 90х годах как стандартная схема для маршрутизации IP адресов.

До CIDR маршрутизаторы в Internet управляли IP трафиком, базируясь исключительно на классах IP адресов и ассоциированных с ними масках подсетей. Такая схема разработки адресного пространства IP неэффективна, как было разъяснено ранее. CIDR указывает более гибкий путь для ассоциации групп IP адресов, не полагаясь на исхоную классовую систему.

12.2.16 Нотация CIDR

CIDR определяет диапазон IP адресов путем уомбинирования IP адреса и ассоциированной с ним сетевой маски. Нотация CIDR использует следующий формат:

```
xxx.xxx.xxx.xxx/n
```

где n — это количество (левостоящих) установленных в '1' бит в маске. К примеру, 192.168.12.0/23 прилагает сетевую маску 255.255.254.0 к сети 192.168, начиная с 192.168.12.0. Такая нотация представляет диапазон адресов 192.168.12.0 – 192.168.13.255. В сравнении с традиционной, базирующейся на классах сетевой нотацией, 192.168.12.0/23 представляет аггрегацию двух сетей класса C 192.168.12.0 и 192.168.13.0, каждая из которых использует маску по умолчанию 255.255.255.0.

CIDR поддерживает выделение адресов Internet и маршрутизацию сообщений независимо от традиционных классов заданного диапазона IP адресов. Например, 10.4.12.0/22 представляет диапазон адресов 10.4.12.0 – 10.4.15.255 путем наложения сетевой маски 255.255.252.0. Так эффективно представляется объединение четырех сетей класса C внутри намного большего пространства класса A.

Нотация CIDR иногда адаптируется даже под не-CIDR сети. В не-CIDR подсетях IP, однако, значения n ограничены либо до 8 (класс A), 16 (класс B), либо до 24 (класс C) из выделения адресов Internet и перспективы маршрутизации.

12.2.17 Как работает CIDR

Гибкость CIDR исходит от доступности маршрутизаторов оперировать с подсетевыми масками, отличными от традиционных масок классов A, B или C (значениея n отличаются от 8, 16 или 24). Для того, чтобы CIDR работала, протоколы маршрутищации Internet должны быть реализованы с поддержкой

соглашений CIDR. Популярные протоколы маршрутизации типа BGP (от англ. Border Gateway Protocol) и OSPF (от англ. Open Shortest Path First) были обновлены для поддержки CIDR несколько лет назад, но менее популярные протоколы все еще не поддерживают CIDR до настоящего момента.

В основном все маршрутизаторы, лежащие в корне Internet (сети WAN между провайдерами), поддерживают CIDR. Основные узлы поддерживают CIDR важным образом для достижения сохранения адресного пространства IP. Частные сети и маленькие публичные LAN менее нуждаются в сохранности адресов, и следовательно могут не утилизировать CIDR.

Для работоспособности подсетей они должны быть непрерывны (расположенные численно рядом) в адресном пространстве. CIDR не способна, к примеру, объединить 192.168.12.0 и 192.168.15.0 в один маршрут без включения промежуточных диапазонов адресов .13 и .14.

12.2.18 CIDR и IPv6

IPv6 обслуживает технологию маршрутизации CIDR и ее нотацию по такому же пути, как и для случая IPv4. IPv6 спроектировано для полного отсутствия классовой адресации.

12.3 Протоколы TCP, UDP, ICMP

Протокол IP используется для передачи пакетов между узлами сети и является транспортным для протоколов UDP, TCP и ICMP. Протокол UDP (User Datagram Protocol) позволяет адресовать пакеты определенным программам узла сети. Протокол TCP (Transmission Control Protocol) позволяет организовать поточный режим передачи между программами узлов сети. Протокол ICMP (Internet Control Message Protocol) в свою очередь используется для передачи сообщений, управляющих работой сети и протоколов высокого уровня.

12.4 Общие сведения о сетевых интерфейсах

Сетевой интерфейс — это элемент операционной системы, предназначенный для взаимодействия между драйвером коммуникационного оборудования и ядром системы. Как ядро системы взаимодействует с сетевым интерфейсом вне зависимости от его типа, так и драйвер взаимодействует с интерфейсом вне зависимости от того, кому предназначаются передаваемые данные и от кого.

Название	Описание
`lo`	Кольцевой интерфейс
`eth0`	Первый интерфейс сети Ethernet
`eth1`	Второй интерфейс сети Ethernet
`ppp0`	Первый интерфейс DialUp PPP

Таблица 12.4: Номенклатура сетевых интерфейсов

Каждый интерфейс определяется названием, IP адресом и маской сети. Таким образом, он однозначно идентифицируется именем внутри системы и IP адресом внутри сети. IP пакеты, предназначенные для определенного адресата, направляются на определенный интерфейс, а пакеты, предназначенные для определенной сети, передаются соответствующему интерфейсу для передачи. Таким образом, когда речь идет о IP адресе узла, всегда имеется в виду IP адрес сетевого интерфейса данного узла.

Название интерфейса определяется типом транспортного протокола и порядковым номером. Исключение составляет кольцевой интерфейс (`loopback`), который является виртуальным интерфейсом и порядкового номера не имеет (Таблица 12.4). Протоколы PPP и Ethernet являются транспортными протоколами для протокола IP, поэтому префикс eth используется для сетей Ethernet (вне зависимости от типа физического носителя), а префикс ppp, соответственно, для соединений PPP (`Point-to-Point Protocol`).

Кольцевой интерфейс lo (`Local LOopback`) присутствует в системе всегда. Он имеет адрес 127.0.0.1, вне зависимости от типа системы и наличия других интерфейсов. В результате, с одной стороны, в системе всегда присутствует хотя бы один сетевой интерфейс, с другой стороны, адрес 127.0.0.1 всегда адресует именно локальную машину.

12.5 Параметры сетевого интерфейса. MTU.

MTU (`Maximum Transmit Unit`) — один из параметров сетевого интерфейса, определяющий максимальный размер пакета, передаваемого через интерфейс. Значение MTU стоит определять в зависимости от пропускной способности интерфейса. Например, передача пакетов большого размера занимает больше времени, при этом остальные пакеты ждут своей очереди, что повышает латентность интерфейса. Если же установить MTU слишком маленьким, это повысит количество служебной информации, передаваемой через интерфейс, и, таким образом, снизит пропускную способность.

Значение MTU обычно устанавливается в пределах между 296 байт для медленных соединений (значение это слагается из размера пакета 256 байт + размера заголовка IP пакета 40 байт) и 1500 байт для локальных сетей.

12.6 Активирование и деактивирование сетевого интерфейса

Настройка, активирование и деактивирование сетевых интерфейсов вручную производится командой ifconfig[2] (`ifconfig lo`):

```
lo Link encap:Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
UP LOOPBACK RUNNING  MTU:16436  Metric:1
RX packets:1569247 errors:0 dropped:0 overruns:0 frame:0
TX packets:1569247 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:3207815212 (3059.2 Mb)  TX bytes:3207815212 (3059.2 Mb)
```

Соответствующий интерфейс активируется командой:

```
ifconfig \
        <название интерфейса> \
        inet <IP адрес> \
        netmask <маска сети> \
        broadcast <широковещательный адрес> \
        up
```

Например, для интерфейса `lo` команда будет выглядеть так:

```
ifconfig \
        lo \
        inet 127.0.0.1 \
        netmask 255.0.0.0 \
        broadcast 127.255.255.255 \
 up
```

Деактивация интерфейса производится командой:

```
ifconfig <название интерфейса> down
```

Выполнение этих команд требует прав доступа администратора. Чтобы не повторяться, добавим, что это относится и ко всем другим командам, фигурирующим в данной главе.

[2] От англ. `InterFace CONFIGurator`.

12.7 Настройка сетевых интерфейсов

Настройки постоянно используемых интерфейсов находятся в специальном каталоге /etc/sysconfig/network-scripts/. Каждый интерфейс определяется файлом с названием ifcfg-<название интерфейса>. Например для интерфейса lo конфигурационный файл имеет особое определенное имя /etc/sysconfig/network-scripts/ifcfg-lo и выглядит следующим образом:

```
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
```

В данном случае интерфейс активируется командой:

```
[root]# ifup lo
```

а деактивируется

```
[root]# ifdown lo
```

Ниже приведен список параметров конфигурационных файлов, используемых для настройки различных интерфейсов, и примеры их значений:

- NAME — название соединения (например, Соединение1);
- DEVICE — название интерфейса (eth0, ppp, lo);
- IPADDR — IP адрес интерфейса;
- NETMASK — маска сети (дается в соответствие с правилами, описанными в разделе 11.2);
- GATEWAY — IP адрес шлюза (как правило адрес находится в той же сети, что и адрес самого интерфейса, например 192.168.1.254 для сети 192.168.1.0/24);
- USERCTL — возможность активирования интерфейса обычным пользователем (принимает значения yes или no);
- MTU — значение MTU для данного интерфейса;

- PEERDNS — включение этой опции предписывает использовать значения серверов DNS, полученных при активировании интерфейса (PPP или DHCP), выключение — при помощи параметров DNS1,2; значения ее — yes или no;
- DNS1, DNS2 — значения первичного и вторичного адресов DNS;
- ONBOOT — включение (yes) режима активирования интерфейса во время загрузки системы;
- BOOTPROTO — указание режима настройки интерфейса; принимает следующие значения: none — при помощи параметров, bootp — при помощи протокола BOOTP, dhcp — при помощи протокола DHCP.

Рассмотрим подробнее настройку интерфейсов Ethernet и PPP.

12.8 Настройка интерфейса Ethernet

Для интерфейса eth0 файл ifcfg-eth0 будет выглядеть следующим образом:

```
NAME=Local Network
DEVICE=eth0
IPADDR=192.168.1.1
NETMASK=255.255.255.0
ONBOOT=yes
```

Если при активировании интерфейса необходимо использовать DHCP, файл ifcfg-eth0 примет следующий вид:

```
NAME=Local Network
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

После редактирования файла конфигурации для интерфейса eth0 его необходимо активировать.

12.8.1 Настройка сетевых интерфейсов при помощи Webmin

Настройка сетевых интерфейсов в Webmin осуществляется при помощи модуля «*Сетевые интерфейсы*», расположенном в подразделе «*Настройка сети*» раздела «*Сеть*». На главной странице модуля показаны список интерфейсов, которые активированы в данный момент и список интерфейсов, которые активируются при запуске системы.

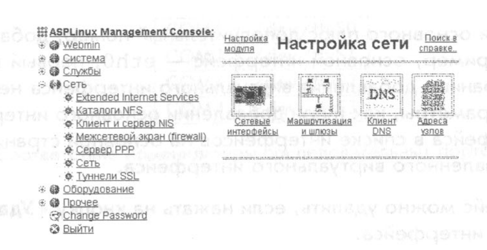


Рис. 12.1: Настройка сети

«Интерфейсы, активируемые при загрузке системы». Все интерфейсы из этого списка имеют соответствующий файл `ifcfg-имя_интерфейса` в директории `/etc/sysconfig/network-scripts`. Но не все из них обязательно включаются при загрузке системы. Информация о том, какие интерфейсы будут подключены, находится в столбце *«Активировать при загрузке?»*.

«Интерфейсы, активные в данный момент» — список работающих интерфейсов. В этом списке могут находиться не только интерфейсы, описанные в предыдущем разделе, но и подключенные пользователем «вручную», а также интерфейсы, подключенные различными специальными программами.

У любого из интерфейсов, показанных в обеих списках, можно изменить параметры, для этого необходимо перейти по ссылке *«название»*. Страница с параметрами у всех интерфейсов одинакова. Единственная дополнительная кнопка «**Сохранить и применить**» будет показана, если редактировать параметры интерфейса, активируемого при загрузке. Для изменения доступны поля: *«IP адрес»*, *«маска сети»*, *«широковещательный адрес»* и *«активировать при загрузке»*.

Добавить новый интерфейс можно в обоих разделах модуля. Для добавления нового интерфейса, активируемого при загрузке, необходимо нажать на ссылке *«Добавить новый интерфейс»*, в нижней части главной страницы модуля. Страница, в которой необходимо ввести параметры нового интерфейса, ничем не отличается от страницы редактирования параметров интерфейса. Если нажать на кнопку «**Создать**», в директории `/etc/sysconfig/network-scripts` будет создан соответствующий интерфейсу файл. Если нажать на кнопку «**Создать и применить**» — файл будет создан, а интерфейс включен.

Если нажать на ссылку *«Добавить новый интерфейс»*, в разделе *«Интерфейсы, активные в данный момент»*, новый интерфейс активируется сразу, но при перезагрузке системы он не будет подключаться, т.к. не создается файл, описывающий его.

Для добавления виртуального интерфейса следует воспользоваться ссылкой *«Добавить виртуальный интерфейс»* на странице редактирования параметров уже существующего интерфейса. Имена виртуальных интерфейсов соот-

ветствуют имени основного плюс дополнительный номер, добавленный через двоеточие. Например, основной интерфейс — eth0, первый виртуальный — eth0:0. На странице добавления виртуального интерфейса необходимо ввести такие же параметры, как и при добавлении основного интерфейса. После создания интерфейса в списке интерфейсов на основной странице модуля появится имя добавленного виртуального интерфейса.

Любой интерфейс можно удалить, если нажать на кнопку «**Удалить**» на странице настройки интерфейса.

Отключить уже активированный интерфейс можно, выбрав параметр «*Неактивен*» на странице редактирования параметров интерфейса. После этого необходимо нажать на кнопку «**Сохранить**». В списке интерфейсов такой интерфейс будет помечен надписью «*Неактивен*». Для включения интерфейса необходимо выбрать параметр «*Активен*» и нажать на кнопку «**Сохранить**».

12.9 Настройка интерфейса PPP

Настройка соединений PPP (DialUp IP) несколько отличается от настройки интерфейсов локальной сети и требует определения следующих дополнительных параметров (в скобках приведены возможные значения):

- PERSIST — включение (yes) или выключение режима восстановления соединения при разрыве связи;
- MODEMPORT — название устройства, к которому подключен модем; в большинстве случаев это один из последовательных портов; номенклатура последних в Linux отличается от таковой в DOS/Windows; так, первый последовательный порт имеет название /dev/ttyS0, и т.д. (соответствие можно определить по таблице 12.5);
- LINESPEED — скорость соединения компьютера с модемом;
- WVDIALSECT — секция в файле конфигурации wvdial, используемая для соединений DialUp, о чем будет сказано ниже;
- DEFROUTE — использование данного соединения в качестве маршрута по умолчанию (возможные значения — yes или no);
- DEBUG — включение (yes) или отключение (no) режима отладки; в первом случае подробный журнал соединения можно видеть в файле /var/log/messages;
- HARDFLOWCTL — включение (yes) или отключение (no) режима аппаратного контроля за передачей данных; при использовании модема эта опция должна быть включена;

COM1	/dev/ttyS0
COM2	/dev/ttyS1
COM3	/dev/ttyS2
COM4	/dev/ttyS3

Таблица 12.5: Соответствие номенклатуры последовательных портов в DOS/Windows и Linux

- PAPNAME — учетное имя, используемое для авторизации;
- DISCONNECTTIMEOUT — количество секунд между разрывом соединения и попыткой восстановления связи; если эта опция не указана, используется значение по умолчанию — 5 секунд;
- RETRYCONNECT — включение/отключение (yes/no) режима автоматического дозвона;
- RETRYTIMEOUT — количество секунд между повторными попытками соединения при автоматическом дозвоне; при отсутствии этой опции используется значение по умолчанию — 60 секунд; разумеется, имеет смысл, только если включен режим автоматического дозвона;
- MAXFAIL — максимальное количество попыток соединения (по умолчанию не установлено);
- DEMAND — включение/отключение (yes/no) режима автоматического соединения при обращении к Internet;
- IDLETIMEOUT — количество времени, через которое будет произведено отключение при отсутствии обмена данными; при отсутствии принимается значение по умолчанию — 600 секунд;
- BOOTTIMEOUT — время, в течение которого система будет ожидать соединения во время загрузки операционной системы; значение по умолчанию — 30 секунд;
- LEASEDLINE — включение (yes) или отключение (no) режима работы с т.н. интеллектуальными модемами, предназначенными для выделенных линий;
- PPPOPTIONS — список дополнительных опций, передаваемых программе pppd, о чем подробнее сказано в интерактивном руководстве Linux — man pppd.

В качестве примера приведем определение интерфейса PPP для выделенной линии. Оно выглядит следующим образом:


```
DEVICE=ppp0
ONBOOT=yes
USERCTL=no
MODEMPORT=/dev/ttyS0
LINESPEED=115200
PERSIST=yes
DEBUG=no
DEFROUTE=yes
HARDFLOWCTL=yes
DISCONNECTTIMEOUT=0
RETRYTIMEOUT=5
BOOTPROTO=none
LEASEDLINE=yes
```

Для настройки соединений DialUp требуется указание дополнительных параметров соединения (в частности номера телефона). В качестве программы автоматической установки соединений DialUp в последнее время широко используется утилита wvdial, которая отрабатывает большинство стандартных ситуаций во время установки связи с провайдером Internet.

Настройка wvdial проводится в две стадии. Первая — исполнение команды

```
[root]# wvdialconf /etc/wvdial.conf
```

в результате чего программа найдет все модемы, подключенные к компьютеру, определит их тип и настройки.

Вторая стадия — добавление в созданный файл /etc/wvdial.conf секции вида:

```
[Dialer ISP]
Username = <учетное имя>
Password = <пароль>
Phone = <телефон доступа>
Inherits = Dialer Defaults
Stupid mode = 1
```

Название секции (в нашем случае это ISP) должно быть указано в соответствующем файле конфигурации интерфейса параметром WVDIALSECT.

При этом если подключение к провайдеру требует некоего сценария подключения, в последней строке следует указать

```
Stupid mode = 0
```

В «умном» режиме wvdial сам ответит на все возможные вопросы, как то: ввод учетного имени, пароля, выбор или ввод протокола подключения. Однако возможно потребуется определить строку, которая будет вводиться при обнаружении неизвестного приглашения, вида

```
Default Reply = <строка>
```

Подробнее о настройках и использовании утилиты wvdial можно прочитать в интерактивном руководстве Linux (man wvdial).

В случае, если после установки соединения производится авторизация средствами PAP или CHAP, необходимо в файлах /etc/ppp/pap-secrets и /etc/ppp/chap-secrets, соответственно, добавить строку, содержащую учетное имя, название интерфейса и пароль:

```
# client         server   secret
<учетное имя>    ppp0     <пароль>
```

12.10 Проверка работоспособности интерфейса

Для проверки работоспособности сети вообще и сетевых интерфейсов (как локальных, так и удаленных) в частности используется команда ping. Эта команда (одна из немногих в этой главе, которая может выполняться от имени обычного пользователя) посылает ICMP-пакеты на указанный интерфейс, который в свою очередь отсылает их обратно, откуда, по ассоциации с пинг-понгом, и происходит ее название.

Приведем пример (вывод команды ping 127.0.0.1):

```
PING 127.0.0.1 (127.0.0.1) from 127.0.0.1 : 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=104 usec
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=67 usec
64 bytes from 127.0.0.1: icmp_seq=2 ttl=255 time=64 usec
64 bytes from 127.0.0.1: icmp_seq=3 ttl=255 time=68 usec
64 bytes from 127.0.0.1: icmp_seq=4 ttl=255 time=92 usec
--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.064/0.079/0.104/0.016 ms
```

В каждой строке этого примера icmp_seq — номер пакета (полезен для визуального определения потери пакетов), ttl — TTL пришедшего пакета (т.н. время жизни пакета), time — время, прошедшее со времени посылки запроса до принятия ответа для определения скоростных характеристик канала передачи данных, измеряемое в следующих единицах: usec — микросекунды, msec — миллисекунды, sec — секунды.

12.10.1 Проверка работоспособности интерфейса при помощи Webmin

Для проверки работоспособности интерфейса в Webmin можно воспользоваться модулем «*Командная оболочка* (shell)», находящемся в разделе

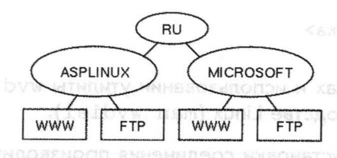


Рис. 12.2: Схема иерархии доменов

«*Прочее*». Этот модуль эмулирует командную строку Linux.

Для выполнения программы **ping**, в поле ввода наберите `ping -c 4 IP_адрес` и нажмите на кнопку «**Выполнить команду:**». Программе **ping** обязательно требуется указать параметр `-c` и количество попыток, так как по умолчанию она будет выполняться до тех пор, пока пользователь не прервет выполнение программы. Если программа **ping** запускается через `Webmin`, пользователь не может прервать выполнение программы (нажав комбинацию клавиш Ctrl+C), следовательно, страница, генерируемая `Webmin`, не будет показана.

12.11 Доменная система имен (DNS)

Как можно видеть из раздела 11.2, для адресации сетевых интерфейсов используются IP адреса в числовой форме. Однако такая система является не совсем удобной с точки зрения конечного пользователя. Поэтому для адресации узлов повсеместно используется доменная система имен (`Domain Name System`, далее DNS).

Доменная система имен (DNS) представляет собой иерархически организованную систему доменов, где каждый домен представляет собой зону ответственности владельца домена за свои поддомены и узлы перед вышестоящим доменом (рис. 12.2).

Доменное имя описывает всю цепь зон ответственности, начиная с нижестоящей. Например, www.asplinux.ru соответствует узлу www (вебсервер) в зоне ответственности asplinux, которая в свою очередь входит в домен ru.

Каждому узлу иерархии может соответствовать один или несколько IP адресов.

База данных соответствий имен и адресов распределена по серверам зон ответственности и поддерживается владельцами доменов. Однако имеется возможность ведения собственной (локальной) базы данных DNS, о чем будет сказано ниже.

12.12 Настройка DNS

В состав ОС Linux входит подсистема поиска имен и адресов узлов, обеспечивающая доступ к базам данных DNS из работающих программ. Ее настройка производится в файлах /etc/hosts.conf и /etc/resolv.conf.

Первый из них, /etc/hosts.conf, — это текстовый файл, определяющий режимы работы подсистемы поиска имен и адресов узлов. Каждая строка должна содержать одно ключевое слово с одним или несколькими параметрами. Рассмотрим подробнее. В строке

```
order p1,p2,p3
```

определяются методы, с помощью которых будет осуществляться поиск IP адреса узла; параметры (p#) могут принимать следующие значения:

- bind — использовать сервер DNS,
- hosts — использовать локальную базу данных,
- nis — использовать NIS.

Параметры (p#) разделяются запятой и указываются в том порядке, в котором будет осуществляться поиск, причем не все три параметра обязательно должны быть указаны.

В строке

```
trim имя_домена
```

ключевое слово может быть использовано несколько раз; в качестве параметра принимается имя домена, начинающееся с точки; при определении имен через сервера DNS, указанный домен будет исключаться из имени.

Строка

```
multi on/off
```

обеспечивает настройку режима обработки локальной базы данных узлов; при включении режима будут учитываться все допустимые адреса узлов, содержащиеся в локальной базе, в противном же случае будет учитываться только первый адрес.

Строка

```
nospoof on/off
```

отвечает за режим проверки подложных имен узлов. При его включении после поиска адреса узла по указанному имени производится поиск имени узла по найденному адресу. Если указанное и найденное имена не совпадают, результат поиска будет признан подложным и игнорируется.

С помощью строки

```
spoofalert on/off
```

включается режим записи результатов проверки подложных имен, определяемой строкой nospoof, в системный журнал.

Строка

```
reorder on/off.
```

включает режим перегруппировки результата поиска таким образом, чтобы локальные адреса были первыми среди найденных.

Не все строки обязательно должны присутствовать в файле /etc/hosts.conf. Например, он может иметь следующий вид:

```
order hosts,bind
multi on
```

Файл /etc/resolv.conf — также текстовый файл, определяющий параметры, используемые подсистемой поиска имен и адресов узлов. Он может содержать строки со следующими значениями:

- nameserver — указание IP адреса сервера DNS, используемого для поиска имен и адресов узлов; может быть указано до трех серверов на случай, если один из них по каким-то причинам будет не доступен; по умолчанию используется адрес локального узла;
- domain — имя локального домена; оно используется при поиске адресов локальных узлов; например, если указан домен asplinux.ru, то при поиске адреса узла www будет произведен поиск адреса узла www.asplinux.ru; как и в предыдущем случае, по умолчанию используется имя домена локального узла;
- search — список доменов для поиска адресов; действие его аналогично ключевому слову domain, за исключением того, что может быть указано несколько доменов, разделенных пробелом, в каждом из которых будет производиться поиск.

Параметры domain и search являются взаимоисключающими. Если встречается несколько таких параметров, учитываться будет только самый последний. Кроме того, действие этих параметров может быть отменено при помощи переменной окружения $LOCALDOMAIN.

Подробнее с параметрами файла /etc/resolv.conf можно ознакомиться в интерактивном руководстве Linux (man resolv.conf).

12.12.1 Настройка клиента DNS при помощи Webmin

Для настройки клиента DNS в Webmin используется модуль *«Клиент DNS»*, находящийся в подразделе *«Сеть»* раздела *«Сеть»*.

В поле *«Имя узла»* следует ввести имя машины. После сохранения изменений, это имя появится в файлах /etc/HOSTNAME и /etc/sysconfig/network. В разделе *«Сервера DNS»* необходимо вписать IP адреса DNS серверов. После сохранения параметров в файле /etc/resolv.conf будут добавлены строки nameserver IP_адрес_сервера. Параметры *«Очередность поиска»* определяют, в каком порядке и к каким системам следует обращаться за преобразованием имени машины в IP адрес. В самом простом случае следует выбрать: hosts, DNS. После изменения эта информация будет сохранена в файле /etc/host.conf.

Если в разделе *«Искать в доменах»* выбрать *«Перечисленных...»* и ввести имена доменов, в файле /etc/resolv.conf появится строка search и указанные домены. Выбор этих параметров позволяет при поиске использовать не FQDN[3] имена машин. Например, в списке указаны домены any.body.com и asplinux.ru. При вводе в командной строке ping www, к имени машины www будет добавлен домен any.body.com и на DNS сервер, для преобразования будет отправлено FQDN имя www.any.body.com. Если DNS не сможет преобразовать такое имя, клиент DNS подставит следующий в списке домен и отправит на преобразование имя www.asplinux.ru.

12.13 Настройка сервера доменной системы имен BIND

BIND (Berkeley Internet Name Domain) представляет собой универсальный сервер DNS. В зависимости от настроек он может выполнять функции главного сервера (хранителя зон), подчиненного и кэширующего сервера.

Главный сервер (master) — сервер, хранящий и обслуживающий зону (домен) DNS.

[3] От англ. Fully Qualified Domain Name — имя, указывающее полный путь к узлу.

Подчиненный сервер (slave) — сервер, не содержащий информацию о домене, но знающий где эта информация находится.

Кэширующий сервер (caching) — сервер, не содержащий информации о доменах, но обрабатывающий запросы клиентов и хранящий результаты обращений к главным серверам для быстрого повторного поиска.

Очень часто сервер DNS выполняет все три функции одновременно, так как он может содержать информацию об одном домене, быть подчиненным сервером другого домена и при этом обслуживать и кэшировать запросы клиентов к остальным доменам.

В данном руководстве мы коснемся настройки только кэширующего сервера, предназначенного для ускорения процесса поиска имен и адресов.

Базовые настройки BIND находятся в файле /etc/named.conf. Он имеет следующий вид:

```
options {
    directory /var/named;
};
zone . IN {
    type hint;
    file named.ca;
};
zone localhost IN {
    type master;
    file localhost.zone;
};
zone 0.0.127.in-addr.arpa IN {
    type master;
    file named.local;
};
```

Данный файл точно определяет местонахождение файлов определений зон (/var/named), отдельную корневую зону, обозначаемую специальным символом . (точка, файл /var/named/named.ca), локальную зону localhost (файл /var/named/localhost.zone), а также обратную локальную зону 0.0.127.in-addr.arpa (/var/named/named.local).

Зона . имеет тип hint и используется для поиска главных серверов доменов. Файл /var/named/named.ca содержит список корневых серверов по всему миру, с которых начинается поиск:

```
. 3600000 S A.ROOT-SERVERS.NET.
. 3600000 NS B.ROOT-SERVERS.NET.
. 3600000 NS C.ROOT-SERVERS.NET
. 3600000 NS D.ROOT-SERVERS.NET.
. 3600000 NS E.ROOT-SERVERS.NET.
. 3600000 NS F.ROOT-SERVERS.NET.
```

```
. 3600000 NS G.ROOT-SERVERS.NET.
. 3600000 NS H.ROOT-SERVERS.NET.
. 3600000 NS I.ROOT-SERVERS.NET.
. 3600000 NS J.ROOT-SERVERS.NET.
. 3600000 NS K.ROOT-SERVERS.NET.
. 3600000 NS L.ROOT-SERVERS.NET.
. 3600000 NS M.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4
B.ROOT-SERVERS.NET. 3600000 A 128.9.0.107
C.ROOT-SERVERS.NET. 3600000 A 192.33.4.12
D.ROOT-SERVERS.NET. 3600000 A 128.8.10.90
E.ROOT-SERVERS.NET. 3600000 A 192.203.230.10
F.ROOT-SERVERS.NET. 3600000 A 192.5.5.241
G.ROOT-SERVERS.NET. 3600000 A 192.112.36.4
H.ROOT-SERVERS.NET. 3600000 A 128.63.2.53
I.ROOT-SERVERS.NET. 3600000 A 192.36.148.17
J.ROOT-SERVERS.NET. 3600000 A 198.41.0.10
K.ROOT-SERVERS.NET. 3600000 A 193.0.14.129
L.ROOT-SERVERS.NET. 3600000 A 198.32.64.12
M.ROOT-SERVERS.NET. 3600000 A 202.12.27.33
```

Число 3600000 определяет время (в секундах), в течение которого корневой сервер будет гарантированно функционировать (1000 часов). И хотя список серверов меняется достаточно редко, его необходимо обновлять время от времени. Свежий список корневых серверов всегда можно получить по адресу `ftp://ftp.rs.internic.net/domain/named.ca`

Зона `localhost` позволяет найти адрес, соответствующий имени localhost — 127.0.0.1, что соответствует локальному кольцевому интерфейсу lo. Определение зоны находится в файле `/var/named/localhost.zone`, имеющем следующий вид:

```
@ IN SOA localhost. root.localhost. (
1997022700 ; Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS localhost.
IN A 127.0.0.1
```

Запись `IN SOA` является началом определения зоны и содержит название (`localhost.`) и, кроме того, адрес администратора зоны, обычно выражаемые как (`root.localhost.` = `root@localhost`). Кроме того, здесь определяются временные характеристики существования зоны.

Запись `IN NS` указывает сервер DNS, отвечающий за данную зону, а запись `IN A` определяет адрес, соответствующий этой зоне.

Зона `0.0.127.in-addr.arpa` является обратной для зоны `localhost`, т.е.

она позволяет определить имя localhost по адресу 127.0.0.1. Определение находится в файле /var/named/named.local:

```
@ IN SOA localhost. root.localhost. (
1997022700 ; Serial
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS localhost.
1 IN PTR localhost.
```

Значение записей IN SOA и IN NS аналогично вышеуказанному, а строка

```
1 IN PTR localhost.
```

определяет имя для адреса 1 в группе 127.0.0.* (т.е. для адреса 127.0.0.1).

Все вышеуказанные файлы находятся в пакете caching-nameserver. Более подробную информацию о настройке BIND можно получить в интерактивной документации Linux (man named.conf).

После произведения изменений в файлах конфигурации BIND необходимо выполнить команду rndc reload для того, чтобы все изменения вступили в силу.

12.13.1 Настройка сервера доменной системы имен BIND при помощи Webmin

Для настройки сервера доменной системы имен BIND в Webmin используется модуль «*Сервер DNS BIND*», находящийся в разделе «*Службы*».

Создание master зоны. Ниже будет рассказано, как создать master зону домена asp-example.net при помощи Webmin. В домене будут находиться машины с именами ns (IP: 1.2.3.4), client1 (IP: 2.3.4.5) и client2 (IP: 3.4.5.6). На машине ns будет работать DNS сервер, отвечающий за зону asp-example.net. На машине client1 будет размещен почтовый сервер. На машине client2 будут размещены www и ftp сервера.

Для создания новой master зоны необходимо выбрать ссылку «*Создать новую зону master*» на главной странице модуля. В появившейся странице укажите, что будет создана зона для прямого преобразования, выберите пункт «*прямая (forward, имена в адреса)*». В разделе «*Имя домена/Сеть*», введите asp-example.net. Файл записей оставьте «*Автоматический*», Webmin самостоятельно выберет имя файла, в котором будет храниться информация о

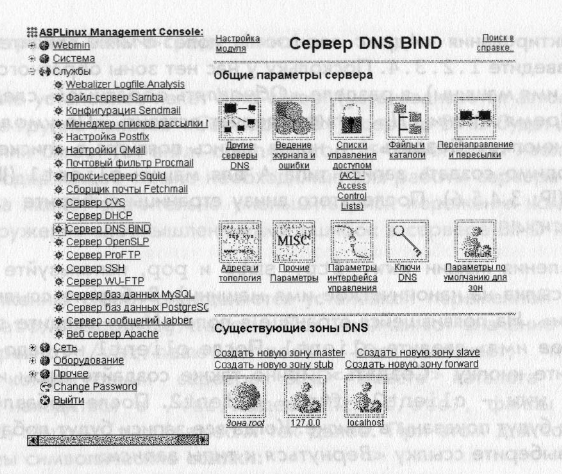


Рис. 12.3: Настройка сервера DNS BIND при помощи Webmin

зоне. По умолчанию этот файл будет находиться в директории /var/named и называться asp-example.net.hosts. В разделе «*Сервер master*» необходимо указать имя master DNS сервера, отвечающего за данную зону, укажите ns.asp-example.net и выберите опцию «*Добавить записи NS для сервера master?*». В поле «*Адрес email*» введите Email человека, ответственного за данный домен. В нашем примере это будет root@asp-example.net.

Поле «*Время обновления*» служит указанием slave серверам, через какое время следует обратиться на master сервер для проверки, были ли внесены изменения в зону. Значение в этом поле рекомендуется установить между 12-ю и 24-мя часами. «*Время повтора передачи*» — необходимо для указания slave серверам, через сколько времени повторить попытку получить информацию о зоне у master сервера, в случае, если предыдущая попытка не удалась. Значение в этом поле можно установить в 15 минут. Slave сервера не будут до бесконечности пытаться получить информацию о зоне, поле «*Время окончания*» служит для указания, через сколько времени после первой неудачной попытки перестать поддерживать зону. Время окончания обычно устанавливают равным 2–3 неделям. В поле «*Время жизни по умолчанию*» указывают, сколько времени информация о записях зоны будет храниться на кэширующих серверах. Обычно это значение равно одному дню. После ввода необходимых параметров нажмите на кнопку «**Создать**».

После создания зоны будет показано окно, в котором можно добавлять новые записи зоны или редактировать уже существующие. На момент создания уже существуют запис и типа SOA и NS.

В первую очередь необходимо добавить записи типа A, служащие для преобразования имен машин в IP адреса. Нажмите на ссылку «*Адрес*». На стра-

нице редактирования *«Адрес записи»* в поле *«Имя»* введите ns, в поле *«Адрес»* введите 1.2.3.4. Поскольку у нас нет зоны обратного преобразования (IP в имя машины), в разделе *«Обновлять обратные?»* следует выбрать *«Нет»*. *«Время жизни TTL»* записи следует оставить *«По умолчанию»*. Нажмите на кнопку *«**Создать**»*, и новая запись появится в списке. Точно также необходимо создать записи типа A для машин client1 (IP: 2.3.4.5) и client2 (IP: 3.4.5.6). После этого внизу страницы выберите *«Вернуться к типы записи»*.

Для добавления машин www, ftp, smtp и pop, используйте записи типа CNAME (ссылка на каноническое имя машины). Выберите ссылку *«Псевдоним имени»*. На появившейся странице в поле *«Имя»* введите smtp, в поле *«Настоящее имя»* введите client1. После client1 не надо ставить точку. Нажмите кнопку «**Создать**». Точно также создайте пары имен: pop – client1, www – client2, ftp – client2. После добавления все новые записи будут показаны в списке. Когда все записи будут добавлены, внизу страницы выберите ссылку *«Вернуться к типы записи»*.

Теперь необходимо указать запись, определяющую машину, которая будет служить почтовым сервером для данного домена. Выберите ссылку *«Почтовый сервер»*. На появившейся странице в поле *«Имя»* введите имя домена asp-example.net., обратите внимание на точку в конце имени домена. В поле *«Почтовый сервер»* введите smtp, без точки в конце имени машины. В поле *«приоритет»* введите целое число, например 5. Нажмите на кнопку *«Создать»*. Вернитесь на главную страницу редактирования зоны.

Для тех, кто привык редактировать файл, описывающий зону «вручную», следует воспользоваться ссылкой *«Редактировать файл записей»*. На странице будет показано поле редактирования, в котором можно исправить соответствующие записи или добавить новые.

Создание slave **зоны.** Для создания slave зоны, в основном окне модуля, следует выбрать ссылку *«Создать новую зону slave»*. На появившейся странице требуется ввести необходимые параметры, описывающие зону. Поле *«Имя домена/Сеть»* должно быть заполнено обязательно, в нем следует вписать имя домена, который будет поддерживаться DNS сервером. В с писке *«Серверы master»* необходимо указать IP адреса master серверов, отвечающих за домен, по одному на строке. Обычно существует только один master сервер. Также необходимо выбрать тип поддерживаемой зоны: прямая или обратная. После ввода всех необходимых параметров, нажмите на кнопку «**Создать**».

Для того, чтобы удалить существующую зону, необходимо выбрать зону и в окне редактирования параметров зоны нажать на кнопку «**Удалить зону**».

12.13.2 Настройка BIND в среде chroot

Если в системе установлен пакет bind-chroot (устанавливается автоматически, если выбрана группа пакетов «*Сервер DNS*»), сервер BIND будет выполняться в окружении chroot (от англ. `change root`). chroot - это изолированное окружение содержащее только необходимые для работы сервера BIND файлы. Эта мера может значительно уменьшить риск поражения компьютера в случае обнаружения злоумышленниками ошибок в сервере BIND и его взлома.

Путь к окружению chroot определяется установкой переменной `$ROOTDIR` в файле `/etc/sysconfig/named`. По умолчанию значение этой переменной равно `/var/named/chroot`. Это означает, что основной файл конфигурации сервера BIND вместо каталога `/etc` теперь будет находиться в `/var/named/chroot/etc/`, файлы зон — в `/var/named/chroot/var/named` и так далее. При этом для совместимости сохранены символические ссылки:

```
/etc/named.conf -> /var/named/chroot/etc/named.conf
/var/named/localdomain.zone -> /var/named/chroot/var/named/localdomain.zone
```

и так далее.

Программа `Webmin` пока не поддерживает bind-chroot и по умолчанию создает файлы зон в каталоге `/var/named`. Поэтому после создания файлов зон программой `Webmin` необходимо их скопировать в каталог `chroot` и создать ссылки для обеспечения возможности последующего редактирования файлов программой `Webmin`. Например, при помощи `Webmin` был создан файл зоны `domain.com.hosts` в каталоге `/var/named`. При использовании bind-chroot нужно выполнить следующие команды:

```
mv /var/named/domain.com.hosts /var/named/chroot/var/named/domain.com.hosts
ln -s /var/named/chroot/var/named/domain.com.hosts /var/named
```

12.14 Настройка локальной базы DNS

Локальная база данных DNS содержится в текстовом файле `/etc/hosts` и используется при отсутствии доступа к серверу DNS (например, во время загрузки системы или в небольших локальных сетях, когда использование сервера DNS не является целесообразным).

Приведем пример файла `/etc/hosts`:

```
127.0.0.1 localhost localhost.localdomain
195.133.213.205 www.asplinux.ru www
```

Каждый IP адрес записывается в отдельной строке в виде:

```
IP_адрес доменное_имя псевдонимы
```

Поля отделяются друг от друга пробелами и/или символами табуляции. Текст, начинающийся с символа #, и до конца строки считается комментарием и игнорируется. Псевдонимы представляют собой измененные, альтернативные, укороченные или обобщенные формы имен узлов.

12.14.1 Настройка локальной базы DNS при помощи Webmin

Для редактирования файла /etc/hosts в Webmin используется модуль «*Адреса узлов*», находящийся в подразделе «*Сеть*», раздела «*Сеть*».

На главной странице модуля показан список — IP адрес — имя машины, а также псевдонимы (aliases) машины. Для добавления новой записи необходимо нажать на ссылке «*Добавить новый адрес узла*». В появившейся странице следует ввести IP адрес и имена машины. После этого необходимо нажать на кнопку «**Создать**». В списке появится новый элемент.

Для редактирования уже существующих записей нажмите на ссылке с IP адресом и в появившейся странице произведите изменения. На той же странице находится кнопка «**Удалить**», при помощи которой запись можно удалить.

12.15 Маршрутизация IP

Маршрутизация IP — это технология, позволяющая определить маршруты IP пакетов, предназначенных определенным адресатам. Под маршрутом IP пакета понимается последовательность узлов сети, через которые проходит пакет на пути от источника к адресату.

Маршрутизация IP может быть определенна статически или же определятся динамически. В данном руководстве мы рассмотрим приемы только статической настройки маршрутов.

Управление таблицей маршрутизации IP в Linux осуществляется при помощи команды route -n в результате чего на экран выводится следующая таблица:

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.2.120  0.0.0.0 255.255.255.255 UH 0 0 0 ppp0
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
0.0.0.0 192.168.2.120 0.0.0.0 UG 0 0 0 ppp0
```

В данной таблице нас интересуют в первую очередь колонки Destination (Пункт назначения), Gateway (Шлюз), Genmask (Маска сети пункта назначения), и Iface (Сетевой интерфейс).

Если передаваемый пакет не предназначается для локальных сетевых интерфейсов, последовательно проверяются правила маршрутизации по данной таблице. Как только встречается правило, при котором адрес маршрутизируемого пакета соответствует пункту назначения найденного правила (с учетом маски сети), пакет направляется на соответствующий сетевой интерфейс для передачи.

Обратите внимание, что последнее правило таблицы маршрутизации имеет маску сети 0.0.0.0 и адрес пункта назначения 0.0.0.0, что соответствует любому IP адресу. Таким образом, пакеты, не удовлетворяющие вышеприведенным правилам, будут пересылаться на адрес 192.168.2.120 через интерфейс ppp0 (шлюз).

В общем случае для компьютера, подключенного к локальной сети, таблица маршрутизации IP будет выглядеть следующим образом:

```
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
0.0.0.0         192.168.1.1 0.0.0.0 UG 0 0 0 eth0
```

Из этой таблицы можно видеть, что все пакеты, предназначенные для локальной сети, будут передаваться через сетевой интерфейс eth0, остальные же пакеты будут передаваться на шлюз сети с адресом 192.168.1.1.

Соответствующие правила маршрутизации пакетов создаются при помощи последовательности команд:

```
[root]# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
[root]# route add default eth0
[root]# route add -net 127.0.0.0 netmask 255.0.0.0 lo
```

Особое внимание следует обратить на то, что для данного примера правила маршрутизации создавать не требуется — правила создаются автоматически при активизации соответствующих интерфейсов.

Дополнительные правила статической маршрутизации IP записываются в файле /etc/sysconfig/network-scripts/route-ethX, где X - номер соответствующего интерфейса:

```
ADDRESS0=10.0.0.0
NETMASK0=255.255.255.0
GATEWAY0=192.168.4.1
ADDRESS1=10.0.1.0
```



```
NETMASK1=255.255.255.0
GATEWAY1=192.168.4.10
```

При активизации какого-либо интерфейса строки из данного файла будут являться параметрами к команде `route`. Подробное описание параметров и использования route дано в интерактивном руководстве Linux (`man route`).

Обращаем особое внимание на то, что задание правил статической маршрутизации является обычно прямым следствием неправильного проектирования сети.

По умолчанию все сетевые интерфейсы принимают пакеты, предназначенные исключительно для данного интерфейса. Для того чтобы сервер Linux выполнял функции маршрутизатора, необходимо разрешить прием всех пакетов, последующая маршрутизация которых будет производиться в соответствии с таблицей маршрутизации IP. Для этого в файл `/etc/sysctl.conf` необходимо добавить строку следующего вида:

```
net.ipv4.ip_forward = 1
```

а затем выполнить команду:

```
[root]# sysctl -p
```

чтобы сделанное изменение вступило в силу.

12.15.1 Управление статической маршрутизацией и шлюзами при помощи Webmin

Для настройки статической маршрутизации и указания шлюзов в Webmin используется модуль «*Маршрутизация и шлюзы*», находящийся в подразделе «*Сеть*» раздела «*Сеть*».

На главной странице модуля можно добавить новые статические маршруты, для этого в разделе «*Статические маршруты*» необходимо ввести название интерфейса, IP адрес сети, маску сети и IP адрес шлюза, через который следует отправлять пакеты. После ввода необходимых параметров следует нажать на кнопку «**Сохранить**». Статический маршрут будет добавлен в файл `/etc/sysconfig/network-scripts/route-<interface>`, где interface — название интерфейса к которому будет привязан заданный маршрут. Изменения в текущую таблицу маршрутизации внесены не будут. Для добавления нового статического маршрута или редактирования параметров уже существующих маршрутов необходимо снова войти в модуль «*Маршрутизация и шлюзы*».

Чтобы удалить существующий маршрут, в разделе «*Статические маршруты*» следует очистить все поля, относящиеся к интересующему маршруту и нажать на кнопку «**Сохранить**».

Если параметр «*Действовать как маршрутизатор*» имеет значение «*Да*», то в файле /etc/sysctl.conf параметру net.ipv4.ip_forward будет присвоено значение 1. Но возможность пересылки пакетов между интерфейсами включена не будет. Необходимо либо перегрузить компьютер, либо воспользоваться модулем «*Командная оболочка (shell)*», расположенном в разделе «*Прочее*». Последнее предпочтительнее. В командной с троке модуля введите команду sysctl -p и нажмите на кнопку «**Выполнить команду:**». «*Шлюз по умолчанию*» предназначен для указания маршрута по умолчанию в таблице маршрутизации. Этот параметр можно получить от DHCP сервера или явно указать IP адрес шлюза.

12.16 Сетевые сервисы

Сетевые сервисы — это программы, позволяющие удаленным пользователям получать доступ к информационным или вычислительным ресурсам сервера. Определенные сервисы обслуживают соединения по определенным портам и протоколам. В файле /etc/services перечисляются названия сервисов, протоколы, посредством которых производится обслуживание, и номера стандартных портов, используемых сетевыми сервисами. Кроме того, в файле определяются альтернативные названия служб.

Все сетевые сервисы по методу обработки запросов делятся на две категории — самостоятельные сервисы и сервисы, подчиненные xinetd.

Самостоятельные сетевые сервисы — это программы-демоны (т.е. программы, запущенные в системе и работающие без участия пользователя), занимающиеся исключительно обработкой сетевых запросов.

Ручное управление самостоятельными сервисами осуществляется при помощи команды service:

```
[root]# service <название сервиса> <команда>
```

где аргумент <команда> может принимать следующие значения:

- start — запуск сервиса,
- stop — остановка сервиса,
- restart — перезапуск сервиса,
- reload — перенастройка сервиса в соответствии с файлами конфигурации.

Список самостоятельных сервисов системы, а также список сетевых сервисов, подчиненных xinetd, можно получить, выполнив команду chkconfig:

```
[root]# chkconfig --list
```

в результате чего будет выведено сообщение вида:

```
crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off
dhcpd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
gpm 0:off 1:off 2:on 3:on 4:on 5:on 6:off
httpd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
iptables 0:off 1:off 2:on 3:on 4:on 5:on 6:off
ircproxy 0:off 1:off 2:off 3:on 4:on 5:on 6:off
keytable 0:off 1:off 2:on 3:on 4:on 5:on 6:off
lpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
named 0:off 1:off 2:off 3:on 4:on 5:on 6:off
netfs 0:off 1:off 2:off 3:on 4:on 5:on 6:off
network 0:off 1:off 2:on 3:on 4:on 5:on 6:off
nfs 0:off 1:off 2:off 3:on 4:on 5:on 6:off
ntpd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
portmap 0:off 1:off 2:off 3:on 4:on 5:on 6:off
sendmail 0:off 1:off 2:on 3:on 4:on 5:on 6:off
smb 0:off 1:off 2:off 3:on 4:on 5:on 6:off
squid 0:off 1:off 2:off 3:on 4:on 5:on 6:off
syslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off
xfs 0:off 1:off 2:on 3:on 4:on 5:on 6:off
xinetd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
xinetd based services:
chargen:        off
chargen-udp:    off
cvspserver:     on
daytime:        off
daytime-udp:    off
echo: off
imap: on
imaps: on
ipop2: on
ipop3: on
ntalk: on
pop3s: on
talk: on
telnet: on
time: on
time-udp:       off
echo-udp:       on
```

Каждая строка верхней таблицы указывает поведение соответствующего сервиса на различных уровнях запуска системы. Нижняя таблица демонстрирует список сетевых сервисов xinetd.

Изменение поведения сервиса производится командой

```
[root]# chkconfig --level <уровни> <название сервиса> <on/off>
```

Например, для исключения сервиса httpd из последовательности загрузки, необходимо выполнить команду:

```
[root]# chkconfig --level 345 httpd off
```

Для включения сервиса httpd в последовательность загрузки следует ввести команду:

```
[root]# chkconfig --level 345 httpd on
```

12.17 Сетевая служба xinetd

Процесс xinetd — это самостоятельная сетевая служба, организующая работу большинства простых сервисов системы. Вместо запуска большого количества самостоятельных служб, предоставляющих сервисы на определенных портах, можно настроить xinetd для приема соединений и запуска программ обработки сервисов.

Сервисы, предоставляемые xinetd, как правило, определяются в файле /etc/xinetd.conf, а также в файлах каталога /etc/xinetd.d/. Файлы определений сервисов распространяются вместе с пакетами, а их структура подробно описана в интерактивном руководстве Linux (man xinetd.conf).

Активирование сетевого сервиса, подчиненного xinetd, также производится при помощи команды chkconfig --level (как это было показано выше). При этом изменения вступают в силу немедленно.

12.17.1 Управление сетевой службой xinetd при помощи Webmin

Для управления сетевой службой xinetd в Webmin используется модуль «*Extended Internet Services*», находящийся в разделе «*Сеть*».

На главной странице модуля показана таблица служб с указанием их параметров. Очень важное значение — разрешена ли служба.

Для редактирования уже существующих служб следует выбрать соответствующую имени службы ссылку. На странице «*Изменение службы интернет*» можно изменить различные параметры. Для ограничения доступа, в параметре «*Разрешить доступ*» необходимо установить «*Только с указанных узлов...*» и добавить в список IP адреса или имена машин, IP адреса или имена сетей, доступ с которых разрешен, по одной записи в строке. Другой способ — это указать машины, доступ с которых запрещен. Также можно

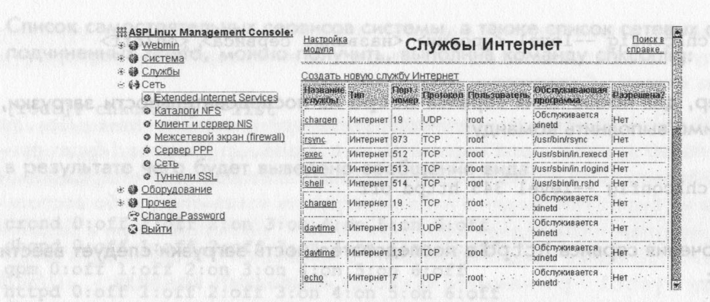


Рис. 12.4: Службы Интернет (xinetd)

указать время, когда клиенты могут подключаться к сервису. Для этих целей служит параметр «*Разрешение доступа в указанное время*». Для ввода временного ограничения в соответствующем поле необходимо набрать данные временного диапазона в формате ЧЧ:ММ–ЧЧ:ММ.

Включение/выключение службы осуществляется выбором параметра «*Обслуживание разрешено?*». Если установлено значение «*Да*» — данная служба будет работать. Если выбрано «*Нет*» — служба работать не будет, но запись о службе останется.

Следующие параметры необходимо указывать в целях защиты от атак на службы: «*Максимальное число одновременно работающих серверов*», «*Макс. число соединений в секунду*», «*Задержка при достижении максимума*».

После внесения изменений следует нажать на кнопку «**Сохранить**». Если службу необходимо удалить из списка служб, нажмите на кнопку «**Удалить**».

Добавить новую службу можно, нажав на ссылку «*Создать новую службу Интернет*». Страница создания службы аналогична странице редактирования параметров. Ввод имени службы является обязательным. Это имя должно быть описано в файле /etc/services или оно должно присутствовать в списке, выводимом модулем «*Сервисы и Протоколы Internet*», находящемся в разделе «*Сеть*». Также, обязательно следует указать путь к программе, которая будет обслуживать подключения к данной службе. Программу указывают в поле «*Обслуживается*». После ввода параметров нажмите на кнопку «**Создать**». Для того, чтобы внесенные изменения вступили в силу, демону xinetd необходимо послать сигнал HUP. Сигнал будет послан после нажатия на кнопку «**Применить изменения**» на главной странице модуля.

12.18 Протокол DHCP

Dynamic Host Configuration Protocol (DHCP) — это протокол динамической настройки узлов сети. Он используется для настройки сетевых интерфейсов клиентских машин во время загрузки операционной системы.

Все настройки сервера DHCP находятся в файле /etc/dhcpd.conf. В общем случае файл должен иметь примерно следующий вид:

```
ddns-update-style none;
subnet 192.168.1.0 netmask 255.255.255.0 {
   option domain-name mydomain.ru;
   option domain-name-servers 192.168.1.1;
   option broadcast-address 195.168.1.255;
   option routers 192.168.1.1;
   host www {
option host-name www;
      fixed-address www.mydomain.ru;
hardware ethernet 00:01:02:58:a5:40;
   }
   host ftp {
option host-name ftp;
      fixed-address ftp.mydomain.ru;
hardware ethernet 00:80:ad:76:98:fb;
   }
   range 192.168.1.10 192.168.1.254;
}
```

Рассмотрим файл конфигурации подробнее. В данном примере определяются параметры локальной сети с адресами 192.168.1.0/255.255.255.0. Серия строк option определяет соответственно имя домена, адрес сервера DNS, широковещательный адрес сети и адрес шлюза сети (о чем говорилось в раздел 11.15 о маршрутизации IP). Далее следует определение фиксированных адресов узлов. Таких определений может сколько угодно, при этом параметры hardware ethernet должны соответствовать внутренним адресам сетевых плат, которые можно узнать при помощи команды ifconfig eth<номер интерфейса>. Команда range определяет область IP адресов выдаваемых другим узлам локальной сети. Т.е. если для интерфейса не определен фиксированный адрес, ему будет выделен любой свободный адрес из диапазона, определяемого минимальным и максимальным адресами включительно.

12.18.1 Конфигурация DHCP сервера при помощи Webmin

Для управления DHCP сервером в Webmin используется модуль «*Сервер DHCP*», находящийся в разделе «*Службы*».

DHCP серверу необходимо указать какие IP адреса в сети будут выделяться клиентам . Если сети не определены, их необходимо добавить. Для этого

следует воспользоваться ссылкой «*Добавить новую подсеть*». В окне «*Создание подсети*» введите IP адрес сети в поле «*Сетевой адрес*», а также соответствующие значения в полях «*Сетевая маска*» и «*Диапазон адресов*». Остальные параметры не обязательны. Затем нажмите на кнопку «**Создать**».

У существующих сетей можно изменить параметры, нажав на ссылке с именем сети. Появится страница с такими же полями, как и при добавлении новой сети. Кроме основных параметров, есть возможность добавления еще одного диапазона адресов, а также кнопки «**Редактирование параметров клиента**», «**Список аренд**» и «**Удалить**». При помощи «*Редактирования параметров клиента*» определяются параметры, передаваемые клиентам выбранной сети. При нажатии на кнопку «**Список аренд**» будет выведена страница со списком аренд, т.е. уже выданными клиентам IP адресами.

Кнопка «**Редактирования параметров клиента**», находящаяся на главной странице модуля, позволяет установить параметры для всех сетей.

На странице, выводимой при нажатии на кнопку «**Редактировать сетевой интерфейс**», можно выбрать сетевые интерфейсы, которые будут обслуживаться DHCP сервером. По умолчанию сервер слушает запросы со всех интерфейсов.

12.19 Система доставки почты sendmail

Программа **sendmail** — это основное средство доставки электронной корреспонденции в Internet. Кроме того, **sendmail** позволяет организовать собственную почтовую службу локальной сети и обмен электронной почтой с другими серверами почтовых служб через почтовые шлюзы.

Правила доставки корреспонденции находятся в файле `/etc/sendmail.cf`, а подробная документация по их конфигурированию — обычно в каталоге `/usr/share/doc/sendmail/` (пакет sendmail-doc). Детальная настройка правил доставки является комплексной задачей и выходит за рамки данного руководства. Однако заметим, что настройка значительно упрощается при использовании пакета sendmail-cf (информацию о котором можно получить в файле `/usr/lib/sendmail-cf/README`).

Обратите внимание, что пакет **sendmail** поставляется настроенным для доставки локальной почты и пересылки исходящей почты на узлы сети в соответствии с определениями DNS. Этого вполне достаточно для организации корпоративной почтовой службы.

Другие параметры, влияющие на доставку корреспонденции, находятся в следующих текстовых файлах: `/etc/aliases`, `/etc/mail/local-host-names`, а также `/etc/mail/virtusertable`. Рассмотрим их подробнее.

Файл `/etc/mail/local-host-names` определяет список локальных поч-

товых доменов. Таким образом, вся почта, предназначенная для указанных доменов, будет доставляться локальным пользователям. Например:

```
company.ru
company.com
company.net
```

После редактирования файла необходимо выполнить команду:

```
[root]# service sendmail reload
```

чтобы сделанные изменения вступили в силу. Файл /etc/aliases определяет список псевдонимов локальных пользователей. Формат списка следующий:

```
<псевдоним>: <адрес1>, <адрес2>, <адрес3>, ...
```

Например, строка

```
manager: ivanov, petrov@another.domain.ru
```

определяет, что вместо доставки письма, адресованного локальному пользователю manager, письмо будет доставлено локальному пользователю ivanov и удаленному — petrov@another.domain.ru.

После произведения изменений в файле /etc/aliases необходимо выполнить команду

```
[root]# newaliases
```

для вступления их в силу.

Файл /etc/mail/virtusertable определяет список виртуальных пользователей. Он является альтернативой файла /etc/aliases, но предоставляет более широкие возможности с точки зрения переадресации сообщений, т.к. влияет не только на входящую, но и на исходящую корреспонденцию. Пример:

```
ivanov@mydomain.ru ivanov@another.domain.ru
petrov@another.domain.ru petrov@mydomain.ru
```

При этом вся входящая и исходящая почта, а также почта, проходящая через данный почтовый узел и адресованная корреспондентам из левой колонки таблицы, будет доставляться по соответствующим новым адресам из правой колонки.

После редактирования файла /etc/mail/virtusertable необходимо выполнить следующие команды

Рис. 12.5: Настройка сервера sendmail при помощи Webmin

```
[root]# cd /etc/mail/
[root]# make
[root]# service sendmail reload
```

после которых изменения вступят в силу.

12.19.1 Настройка сервера sendmail при помощи Webmin

Для настройки сервера **sendmail** в Webmin используется модуль «*Конфигурация Sendmail*», находящийся в разделе «*Службы*».

Самая простейшая настройка **sendmail**, предусматривает разрешение отправки почты через сервер для клиентов локальной сети, а также указание имени домена, для пользователей которого будет приниматься почта.

По умолчанию **sendmail** разрешает отправлять почту только локальным пользователям. Для указания, кому можно использовать **sendmail** для отправки почты, следует выбрать ссылку «*Spam Контроль(access)*». На странице «*Spam контроль*» показан список имен машин, IP адресов компьютеров или сетей. Если напротив указанных в списке машин или сетей присутствует надпись RELAY, значит клиентам, находящимся на этих машинах или в этих сетях, разрешено отправлять почту при помощи **sendmail**.

Для добавления новой машины в список, в области «*создание Правил Spam Контроля*» можно выбрать источник, от которого будет приниматься почта: почтовый адрес, сеть, пользователь, домен. При помощи spam контроля можно не только указывать, кто будет отправлять почту через сервер, но и некоторые дополнительные возможности. Если необходимо разрешить именно пересылку почты (relay), то в качестве источника следует выбирать только сеть или домен. Например, нам надо разрешить принимать почту от машины с IP адресом 192.168.2.3, в источнике следует выбрать «*сеть*», а в поле ввода написать IP адрес. В опциях действий выберите «*Разрешить трансляцию*» и нажмите на кнопку «**Создать**». В списке появится IP адрес машины и

действие RELAY. Если машина, на которой работает **sendmail**, имеет доступ к DNS серверу или соответствие имени машины IP адресу описано в файле /etc/hosts, для разрешения пересылки почты в качестве источника можно выбирать «*Домен*» и указывать имя машины. Для удаления машины из списка следует выбрать имя или IP адрес машины в списке и на появившейся странице нажать на кнопку «**Удалить**».

Иногда проще вручную редактировать файл доступа. Следует выбрать ссылку «*Manually edit /etc/mail/access*». На странице будет показано поле, в котором находится содержимое файла. Одно правило записывается на одной строке. Для того, чтобы добавить IP адрес компьютера, следует в начале строки вписать его IP адрес или имя, затем через пробел или табуляцию написать ключевое слово RELAY. Не забывайте в конце файла оставлять одну пустую строку. Затем нажмите на кнопку «**Сохранить**» и изменения вступят в силу.

Для разрешения доступа клиентов к почтовому серверу, необходимо обязательно изменить опцию «*SMTP port options*», находящуюся на странице «*Параметры Sendmail (O)*» . Напротив этого параметра следует выбрать «*По умолчанию*». Затем нажмите на кнопку «**Сохранить и Активизировать**». Если эта опция не будет изменена — **sendmail** никогда не будет принимать почту для пересылки с других машин, даже если пересылка разрешена в файле /etc/mail/access.

Sendmail принимает почту только для домена, совпадающего с именем компьютера. Предположим, что имя компьютера smtp.asp-example.net и на нем заведен пользователь sample. Почта, которую отослали по адресу sample@smtp.asp-example.net будет доставлена в почтовый ящик пользователя sample. Если в DNS сервере в описании зоны asp-example.net присутствует запись MX, ссылающаяся на машину smtp.asp-example.net, **sendmail** на этой машине не примет почту для sample@asp-example.net. Для того, чтобы он начал принимать почту для домена asp-example.net на главной странице модуля «*Конфигурация Sendmail*», выберите ссылку «*Локальные домены (Cw)*». В поле редактирования, на новой строке введите имя домена asp-example.net и нажмите на кнопку «**Сохранить**».

Для работы с псевдонимами необходимо выбрать ссылку «*Почтовые Псевдонимы (aliases)*». На странице будет показан список уже существующих псевдонимов, напротив которых указан пользователь, которому будет пересылаться почтовое сообщение. Для создания нового псевдонима, его имя необходимо ввести в поле «*Адрес*», поле «*Активен?*» должно быть установлено в значение «*Да*». В списке «*Псевдоним к*» следует выбрать тип и в поле ввода ввести необходимое значение. Например, необходимо создать псевдоним sample, ссылающийся на пользователей root и ftp. Для этого в поле «*Адрес*» введите sample, выберите «*Да*», установите «*Псевдоним к*» в «*Почтовому адресу*», а в поле ввода наберите root,ftp. Затем нажмите кнопку «**Создать**».

Другой способ добавления псевдонимов — вручную отредактировать файл /etc/aliases. Внизу окна выберите ссылку *«Manually edit /etc/aliases»*. На новой странице будет показано поле редактирования, в котором находится содержимое файла /etc/aliases. Для добавления псевдонима, в новой строке следует ввести имя псевдонима, заканчивающееся двоеточием. Затем, после пробела или табуляции указать имена пользователей, разделенных запятыми.

```
sample: root,ftp
```

Для сохранения изменений, нажмите на кнопку «**Сохранить**». Теперь почта, направляемая пользователю sample, будет пересылаться как пользователю root, так и пользователю ftp. После всех изменений, которые были произведены с параметрами **sendmail**, его необходимо будет перезапустить. На главной странице модуля следует нажать на кнопку *«Остановить sendmail»*. После перерисовки страницы нажмите на кнопку *«Запустить sendmail»*.

12.20 Почтовые сервисы POP3 и IMAP

В качестве основного сервера IMAP/POP3 в дистрибутиве **ASPLinux** используется пакет dovecot. dovecot - это IMAP/POP3 сервер с открытыми исходными текстами для Linux/UNIX, при написании которого особое внимание уделялось вопросам безопасности. В качестве базы сообщений dovecot использует стандартные форматы mbox и maildir.

12.20.1 Установка и настройка пакета

Для установки сервера IMAP/POP3 необходимо установить пакет dovecot и все необходимые для него по зависимостям пакеты.

По умолчанию в пакете dovecot разрешены только протоколы imap и imaps. Для включения поддержки pop3 и pop3s необходимо изменить файл конфигурации /etc/dovecot.conf. В нем необходимо добавить строку:

```
protocols= imap imaps pop3 pop3s
```

После установки пакета автоматический запуск службы **dovecot** отключен. Чтобы включить автоматический запуск dovecot, нужно выполнить команду

```
chkconfig dovecot on
```

или воспользоваться утилитами **ntsysv** или **system-config-services**. Для запуска службы **dovecot** вручную введите команду

```
service dovecot start
```

12.20.2 Проверка работы сервера POP3

Зная имя и пароль одного из пользователей системы, можно проверить работу сервера следующим образом:

```
telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
+OK dovecot ready.
user userXX
+OK
pass XXXX
+OK Logged in.
stat
+OK 1 864
quit
+OK Logging out.
```

12.20.3 Поддержка SSL

В составе пакета dovecot поставляется демонстрационный сертификат для адреса localhost.localdomain. Сертификат находится в файле /usr/share/ssl/certs/dovecot.pem. Рекомендуется создать свой сертификат. Для этого необходимо выполнить следующую команду:

```
cd /usr/share/ssl
openssl req -new -x509 -days 365 -nodes \
    -out certs/dovecot.pem \
    -keyout private/dovecot.pem
```

Введите ответы на вопросы, которые последуют после этой команды. Просмотреть полученный сертификат можно будет командой

```
openssl x509 -noout -subject  < /usr/share/ssl/certs/dovecot.pem
```

12.21 Web-сервер Apache

В качестве Web-сервера в Linux широко используется программа Apache. Она выступает в качестве самостоятельного сетевого сервиса и поддерживает средства CGI, SSL, язык PHP и многое другое.

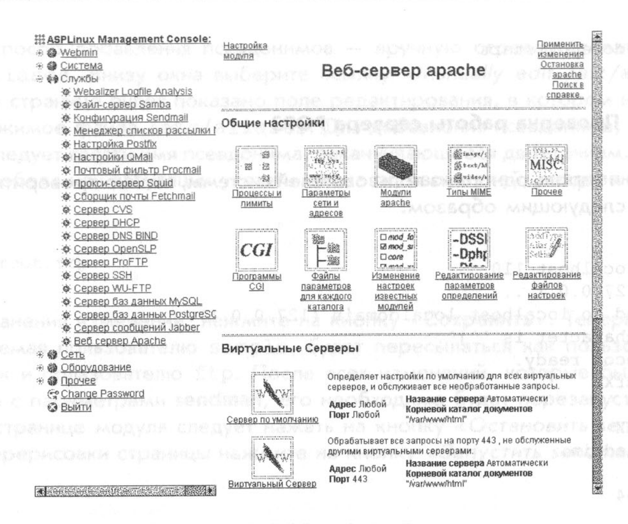


Рис. 12.6: Настройка WEB сервера Apache при помощи Webmin

Все настройки сервера находятся в каталоге /etc/httpd/conf/. В нем определяются корневой каталог Web-сервера (/var/www/html/), далее, каталог скриптов CGI (/var/www/cgi-bin/), набор различных подключаемых модулей и другие параметры.

Сервер полностью настроен и работоспособен сразу после установки. В этом можно убедится, открыв в любом браузере страницу http://localhost/, например, таким образом:

```
lynx http://localhost/
```

Подробное руководство по настройке Apache, а также описание подключаемых модулей находится в пакете apache-manual. Вся эта документация сконцентрирована в каталоге /var/www/html/manual/ и настроена так, что становится доступна (через браузер) по адресу http://localhost/manual/

12.21.1 Настройка WEB сервера Apache при помощи Webmin

Для настройки WEB сервера Apache в Webmin применяется модуль «*Веб сервер Apache*», находящийся в разделе «*Службы*».

В большинстве случаев сервер Apache настраивать не требуется. На что следует обратить внимание при настройках сервера — это ссылка «*Процессы и лимиты*». На этой странице описываются различные ограничения, на которые

следует обратить внимание: *«Максимальное число одновременных запросов»*, *«Максимальное число запросов на процессы сервера»* и *«Максимальное число зарезервированных процессов сервера»*. Если сервер не справляется с количеством запросов, эти параметры следует изменить.

Иногда необходимо, чтобы Apache кроме основного сервера обслуживал еще несколько виртуальных. Хотя Apache и позволяет использовать виртуальный хостинг, базирующийся на IP адресах, с появлением протокола HTTP версии 1.1 в основном используется хостинг, базирующийся на именах WEB серверов. Для добавления виртуального WEB сервера, внизу главной страницы модуля в поле *«Адрес»* следует выбрать *«любой»*, *«Порт»* можно оставить *«По умолчанию»*. В поле *«Корневой каталог»* необходимо выбрать директорию, где будут находиться HTML страницы и прочие файлы WEB сервера. Обязательно следует указать *«Название сервера»*. После ввода параметров нажмите на кнопку **«Создать»**, виртуальный сервер появится в списке серверов, обслуживаемых Apache. Более детальное конфигурирование виртуального сервера можно осуществить на странице *«Параметры виртуального сервера»*.

Для того, чтобы изменения вступили в силу, вверху страницы модуля *«Веб сервер Apache»* существует ссылка *«Применить изменения»*. Эта ссылка присутствует только тогда, когда WEB сервер запущен.

12.22 Прокси-сервер SQUID

SQUID — это высокопроизводительный кэширующий прокси-сервер для Web-клиентов, поддерживающий протоколы HTTP, FTP и Gopher.

Настройки SQUID находятся в файле `/etc/squid/squid.conf`. Причем, с одной стороны, файл достаточно хорошо документирован: к каждой опции прилагаются обширные комментарии, с другой — вносить изменения в конфигурацию, как правило, не требуется: сервер работоспособен с настройками по умолчанию.

Поэтому ниже будут рассмотрены некоторые опции, изменение которых администратором конкретной системы наиболее вероятно. Среди таких опций строка

```
http_port 3128
```

определяет порт TCP/IP, на котором SQUID принимает соединения клиентов. Строка

```
icp_port 3130
```

указывает на порт UDP/IP, через который производится межсерверный обмен данными в соответствии с протоколом ICP (RFC2186 и RFC2187). Опция

```
cache_peer <имя узла> <тип> <порт HTTP> <порт ICP>
```

определяет вышестоящие и одноранговые прокси-серверы. Одноранговые прокси-серверы удобны для организации кластеров из прокси-серверов. Одноранговый прокси-сервер определяется строкой:

```
cache_peer <имя узла> sibling <порт HTTP> <порт ICP>
```

Настройка на вышестоящий прокси-сервер (например, провайдера) производится в следующей строке:

```
cache_peer <имя узла> parent <порт HTTP> <порт ICP>
```

Если вышестоящий сервер не поддерживает протокол ICP, следует для порта ICP установить значение, равное 7 (UDP echo request).

В строке

```
cache_mem 8 MB
```

определяется количество памяти, используемой прокси-сервером для кэширования. Увеличение этого значения повышает производительность прокси-сервера, уменьшение же приведет к освобождению оперативной памяти для использования в других целях.

С помощью опции

```
cache_dir ufs /var/spool/squid 100 16 256
```

определяется тип, местоположение, размер и параметры дискового кэша. В данной опции можно беспрепятственно менять только размер дискового кэша (в приведенном примере он равен 100 Мбайт). После изменения других параметров необходимо заново инициализировать дисковый кэш, для чего служит команда

```
squid -z
```

в консольном режиме. Две взаимосвязанные строки,

```
acl QUERY urlpath_regex cgi-bin \?
```

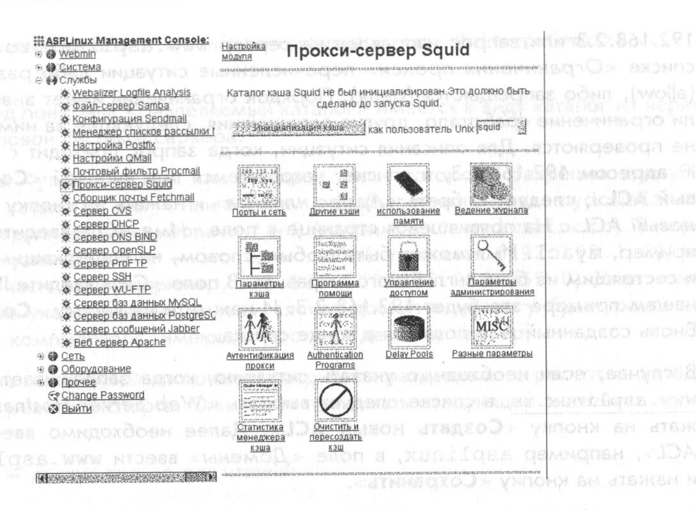


Рис. 12.7: Настройка прокси-сервера SQUID при помощи Webmin

и

```
no_cache deny QUERY
```

указывают список регулярных выражений, определяющих страницы, которые не будут кэшироваться. В приведенном примере это все URL, содержащие cgi-bin или ?.

После любых изменений, внесенных в файл /etc/squid/squid.conf необходимо перезапустить SQUID при помощи команды:

```
[root]# service squid restart
```

Лишь после этого изменения вступят в силу.

12.22.1 Настройка прокси-сервера SQUID при помощи Webmin

Для настройки прокси сервера SQUID в Webmin применяется модуль *«Прокси сервер SQUID»*, находящийся в разделе *«Службы»*.

Ограничения доступа к серверу происходит на странице, вызываемой при использовании ссылки *«Управление доступом»*. Ограничения вводятся следующим способом: сначала описывается некоторая ситуация (ACL) в списках управления доступом. Например, запрос пришел с машины с IP адресом

192.168.2.3 или запрос направлен на сервер www.asplinux.ru. Затем, в списке *«Ограничения прокси»* перечисленные ситуации либо разрешаются (allow), либо запрещаются (deny). Порядок ограничений имеет значение, если ограничение сработало, другие ограничения, следующие за ним в списке, не проверяются. Для описания ситуации, когда запрос приходит с машины с IP адресом 192.168.2.3, в списке, находящемся под кнопкой «**Создать новый ACL**», следует выбрать *«Адрес клиента»* и нажать на кнопку *«Создать новый ACL»*. На появившейся странице в поле *«Имя ACL»* введите имя, например, myacl. Имя может быть любым словом, не содержащим пробелы и состоящим из букв английского алфавита. В поле *«CIP»* введите IP адрес, в нашем примере это будет 192.168.2.3. И нажмите на кнопку «**Сохранить**». Вновь созданный ACL появится в конце списка.

В случае, если необходимо указать ситуацию, когда запрашивается ресурс www.asplinux.ru, в списке следует выбрать *«Web Server Hostname»* и нажать на кнопку «**Создать новый ACL**». Далее необходимо ввести *«Имя ACL»*, например asplinux, в поле *«Домены»* ввести www.asplinux.ru и нажать на кнопку «**Сохранить**».

Для того, чтобы запретить доступ к WEB серверу www.asplinux.ru с компьютера 192.168.2.3, выберите ссылку *«Добавить прокси ограничение»*. На следующей странице необходимо выбрать *«Запретить»*. В списке *«Совпад. с ACL»*, выберите myacl и удерживая клавишу Ctrl, выберите asp. Нажмите на кнопку «**Сохранить**». В списке *«Ограничения прокси»*, самым последним ограничением должно быть *«Deny all»*. Для того, чтобы внесенное ограничение сработало, его следует переместить на необходимую позицию, нажимая на стрелки.

Удалить ограничение можно, нажав на ссылку с его именем и на появившейся странице нажать на кнопку «**Удалить**». Аналогично удаляются ACL.

Чтобы изменения вступили в силу при работающем прокси– сервере, нет необходимости его перезапускать. Достаточно выбрать ссылку *«Принять изменения»*, находящуюся вверху главной страницы модуля.

12.23 Сетевая файловая система NFS

Для разделения дискового пространства на серверах под управлением операционных систем семейства UNIX/Linux широко применяется NFS (Network File System — Сетевая Файловая Система). Она предоставляет отдельные каталоги из иерархии файловой системы сервера для чтения или чтения и записи клиентами NFS.

Настройка разделяемых ресурсов осуществляется в особом текстовом файле /etc/exports, в котором каждая строка определяет один ресурс и имеет следующий вид:

```
<разделяемый каталог> <узел1>(<атрибуты1>) <узел2>(<атрибуты2>) ...
```

Здесь под понятием <разделяемый каталог> имеется в виду каталог из иерархии файловой системы сервера.

Значение <узел> — имя или IP адрес узла, которому разрешен доступ к данному каталогу; имя — или в явном виде (например, host.mydomain.ru) или — с групповыми символами * и ? (например, *.mydomain.ru). В определении узла IP адрес — это один любой узел сети (192.168.1.2), вся сеть или часть сети (например, 192.168.1.0/255.255.255.0). Узел может быть и не указан совсем; в этом случае доступ к каталогу будет разрешен для всех компьютеров, имеющих доступ к данному серверу.

В понятие <атрибуты> включаются атрибуты разделяемого ресурса, определяющие режим экспорта. Они могут принимать следующие значения:

- ro — доступ только для чтения;
- rw — доступ для чтения и записи;
- sync — устанавливает синхронный режим записи, при котором все данные будут передаваться серверу до окончания команды записи;
- async — устанавливает асинхронный режим записи, при котором данные будут передаваться серверу по мере его готовности;
- wdelay — предписывает производить задержку записи после получения данных на запись; это позволяет повысить производительность при частом изменении одних и тех же файлов;
- no_wdelay — устанавливает режим немедленной записи данных на диск;
- root_squash — указывает, что операции, выполняемые от имени администратора системы на клиентах NFS, будут выполняться на сервере от имени анонимного пользователя;
- no_root_squash — указывает, что операции, выполняемые от имени администратора системы на клиентах NFS, будут выполняться от его же имени также и на сервере; данная опция крайне небезопасна, однако полезна для организации бездисковых станций;
- all_squash — указывает, что все операции на сервере будут производиться только от имени анонимного пользователя;
- no_all_squash — указывает, что все операции на сервере будут производиться с использованием идентификаторов пользователя и группы клиента; в данном случае необходимо убедиться, что все пользователи имеют одинаковые идентификаторы на всех компьютерах сети;

- anonuid=<uid> и anongid=<gid> — атрибуты, определяющие идентификатор анонимного пользователя и идентификатор анонимной группы, от которых будут производиться анонимные операции (см. выше, в описании атрибутов root_squash и all_squash).

Атрибуты разделяются запятыми. При их отсутствии действуют значения по умолчанию (ro, async, wdelay, no_all_squash, anonuid=65534,anongid=65534). Описание атрибутов есть в интерактивном руководстве Linux (man exports). Приведем пример файла /etc/exports:

```
/home/ftp *.mydomain.ru(ro)
/opt/projects proj*.mydomain.ru(rw)
/home/joe pc001.mydomain.ru(rw,all_squash,anonuid=150,anongid=150)
/opt/public (ro,all_squash)
```

После произведения изменений в файле /etc/exports, для того чтобы изменения вступили в силу, необходимо выполнить команду

```
exportfs -r
```

подробности о которой можно узнать из man exports.

Подключение разделяемых каталогов на клиентских машинах производится при помощи команды mount:

```
mount <имя сервера>:<разделяемый каталог> <место подключения>
```

Например, это можно выполнить следующим образом:

```
mount server.mydomain.ru:/opt/projects /mnt/projects
```

Для автоматического монтирования разделяемых ресурсов во время загрузки операционной системы необходимо добавить в файл /etc/fstab строку вида:

```
<имя сервера>:<разделяемый каталог> <место подключения> nfs <атрибуты>
```

например, таким образом:

```
server.mydomain.ru:/opt/projects /mnt/projects nfs defaults
```

Отключение разделяемых ресурсов осуществляется командой umount, аналогично тому, как это делается для локальных файловых систем.

12.23.1 Настройка сервера NFS при помощи Webmin

Для настройки сервера NFS в Webmin используется модуль «*Каталоги NFS*», находящийся в разделе «*Сеть*».

После установки **ASPLinux** не определено ни одного экспортируемого каталога, которые могут быть подключены клиентами в сети. Чтобы экспортировать каталог, необходимо нажать на ссылку «*Добавить каталог для экспорта*». В поле «*Экспортируемый каталог*» введите путь к каталогу, указанный каталог уже должен существовать. В поле «*Включить?*» выберите «*Да*». Затем необходимо выбрать, с какой машины можно подключать этот каталог. Обычно указывают имена, IP адреса машин или IP адреса сетей. «*Режим доступа*» определяет, доступен ли каталог только для чтения или для чтения и записи. И еще один важный параметр — «*Доверять удаленным пользователям*». Обычно выбирают «*Всем, кроме root*», эта опция аналогична указанию `root_squash` в опциях экспорта в файле `/etc/exports`.

Если необходимо, чтобы все создаваемые файлы в экспортируемой директории принадлежали пользователю `nfsnobody`, следует выбрать «*Никому*», эта опция аналогична `all_squash`. После указания всех необходимых параметров нажмите на кнопку «**Создать**». Если были добавлены новые экспортируемые директории, или у существующих директорий были изменены параметры экспортирования, необходимо воспользоваться кнопкой «**Применить изменения**».

12.24 Сетевой экран

Кроме обеспечения доступа к различным сервисам, в задачи администратора сети входит ограничение доступа к определенным службам и узлам, т.е. организация сетевого экрана.

Сетевой экран — это средство управления доступом к определенному участку сети, узлу или сервису. В функции сетевого экрана входит:

- фильтрация пакетов, не удовлетворяющих определенным условиям;
- трансляция IP адресов и/или портов с целью перенаправления трафика на другие узлы и маскирования узлов локальной сети;
- ведение журнала и сбор статистики.

Сетевой экран обычно располагается в местах соединения сетей и экранирует сети друг от друга.

Функции сетевого экрана в Linux выполняет подсистема `iptables`, входящая в ядро Linux версии 2.6.x, используемое в дистрибутиве **ASPLinux**. Эта под-

система представляет собой набор функциональных модулей сетевого экрана, поведение каждого из которых определяется набором правил, сгруппированных в блоки последовательных правил (цепочек). Кроме того, каждая цепочка имеет собственную политику, определяющую правило обращения с IP-пакетом, не удовлетворяющим ни одному из указанных условий.

Настройка правил экранирования осуществляется при помощи одноименной команды `iptables`.

Подробное описание опций команды `iptables` дано в интерактивном руководстве Linux (`man iptables`). В настоящем же руководстве будет рассмотрен достаточно типичный пример настройки сетевого экрана. Допустим, у нас имеется небольшая локальная сеть с адресами 192.168.0.* и сервер, подключенный к Internet. На сервере установлены службы DNS, sendmail, apache, squid и pop3. Определим, что eth0 — интерфейс локальной сети сервера с адресом 192.168.0.1, а ppp0 — сетевой интерфейс Internet с адресом 194.236.50.155. Необходимо решить следующие задачи:

- настроить трансляцию адресов так, чтобы пользователи сети имели доступ в Internet;
- экранировать ресурсы локальной сети и сервера, оставив доступными только почтовый сервис и сервис Web — apache.

Последовательность команд, определяющих сетевой экран, в нашем случае будет следующей:

```
iptables -P INPUT DROP
iptables -A INPUT -p ALL -i eth0 -s 192.168.0.0/255.255.255.0 -j ACCEPT
iptables -A INPUT -p ALL -d 127.0.0.1 -j ACCEPT
iptables -A INPUT -p ALL -d 192.168.0.1 -j ACCEPT
iptables -A INPUT -p TCP -d 194.236.50.155 --dport smtp -j ACCEPT
iptables -A INPUT -p TCP -d 194.236.50.155 --dport www -j ACCEPT
iptables -A INPUT -p UDP -s 0/0 --source-port domain -j ACCEPT
iptables -A INPUT -p UDP -d 0/0 --dport domain -j ACCEPT
iptables -A INPUT -p ICMP -s 0/0 --icmp-type echo-reply -j ACCEPT
iptables -A INPUT -p ICMP -s 0/0 --icmp-type destination-unreachable -j ACCEPT
iptables -A INPUT -p ICMP -s 0/0 --icmp-type redirect -j ACCEPT
iptables -A INPUT -p ICMP -s 0/0 --icmp-type time-exceeded -j ACCEPT
iptables -P FORWARD DROP
iptables -A FORWARD -i eth0 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -P OUTPUT DROP
iptables -A OUTPUT -p ALL -s 127.0.0.1 -j ACCEPT
iptables -A OUTPUT -p ALL -s 192.168.0.1 -j ACCEPT
iptables -A OUTPUT -p ALL -s 194.236.50.155 -j ACCEPT
```

Рис. 12.8: Настройка сетевого экрана при помощи Webmin

```
iptables -t nat -A PREROUTING -i ppp0 -s 192.168.0.0/255.255.255.0 -j DROP
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

После завершения настройки сетевого экрана, все правила необходимо сохранить при помощи команды `service iptables save`.

В результате ее во время последующей загрузки системы все правила автоматически восстановятся.

12.24.1 Настройка сетевого экрана при помощи Webmin

Для настройки сетевого экрана с помощью Webmin используется модуль «*Межсетевой экран Linux (firewall)*», по умолчанию находящийся в разделе «*Сеть*».

Если в системе не существует файла /etc/sysconfig/iptables, на первой странице модуля предлагается сформировать его. Выберите «*Allow all traffic*» и нажмите на кнопку «**Setup Firewall**». Для того, чтобы создать такие же настройки как и в предыдущем разделе, при помощи Webmin, необходимо выполнить следующие действия: Сначала установим политики по умолчанию для цепочек INPUT, FORWARD и OUTPUT. Их необходимо изменить на DROP. На странице есть три раздела, показывающие какие правила фильтрации определены в цепочках. Для установки политик по умолчанию необходимо в списках, находящихся возле кнопок «*Действие по умолчанию*», выбрать

«*Отбрасывать*» и нажать на соответствующую кнопку. Затем создадим правила фильтрации на цепочке INPUT. Для добавления правила следу ет нажать на кнопку «**Add rule**» в разделе «*Входящие пакеты (INPUT)*». Правило, которое будет добавлено, выглядит следующим образом:

```
iptables -A INPUT -p ALL -i eth0 -s 192.168.0.0/24 -j ACCEPT
```

В поле «*Действие*» выберите «*Принимать*». В поле «*Адрес или сеть источника*» необходимо выбрать «*Равно*» и ввести 192.168.0.0/24. «*Входящий интерфейс*» — в списке выберите «*Равно*». Далее, в списке интерфейсов необходимо выбрать «*eth0*». При описании правил фильтрации можно указывать несуществующие интерфейсы, в случае выбора такого интерфейса, правила будут применяться тогда, когда этот интерфейс будет включен. К таким интерфейсам относятся ppp0 и подобные. Если будет описываться еще невключенный интерфейс, в списке интерфейсов следует выбрать «*Другой..*», а в поле ввода вписать необходимый интерфейс. После ввода всех правил фильтрации необходимо нажать на кнопку «**Создать**». Новое правило появится в списке правил цепочки INPUT.

Следующие два правила добавляются аналогичным образом, только при вводе правил не указывается входной интерфейс, а вместо «*Адрес или сеть источника*», выбирают «*Адрес или сеть назначения*».

```
iptables -A INPUT -p ALL -d 127.0.0.1 -j ACCEPT
iptables -A INPUT -p ALL -d 192.168.0.1 -j ACCEPT
```

Добавление правил, где указывается тип протокола и порт назначения, осуществляется следующим образом. Добавим правило:

```
iptables -A INPUT -p TCP -d 194.236.50.155 --dport smtp -j ACCEPT
```

«*Действие*» установите в «*Принимать*». «*Адрес или сеть назначения*» — в списке выберите «*Равно*», а в поле ввода напишите 194.236.50.155. «*Сетевой протокол*» — установите в «*Равно*», а в списке выберите TCP. «*Порт TCP или UDP назначения*» — выберите «*Равно*», установите «*Порт(ы)*» и введите `smtp` или 25. Нажмите на кнопку «**Создать**».

Остальные правила в цепочках INPUT, OUTPUT и FORWARD добавляются аналогично. Правила

```
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
iptables -t nat -A PREROUTING -i ppp0 -s 192.168.0.0/255.255.255.0 -j DROP
```

добавляются в таблице `nat`. Чтобы работать с этой таблицей, вверху страницы, в списке, находящемся возле кнопки «**Showing IPtable:**», выберите «*Преобразование сетевых адресов (nat)*» и нажмите на кнопку. Как и в предыдущих примерах, для добавления правила необходимо нажать на кнопку «**Добавить правило**», в разделе «*Пакеты после маршрутизации (POSTROUTING)*». «*Действие*» — установите в «*Маскировать*». «*Исходящий интерфейс*» — выберите «*Равно*», «*Другой*» и введите ppp0. Нажмите

на кнопку «**Создать**». Новое правило будет показано в списке правил цепочки POSTROUTING. Второе правило добавляется в цепочку PREROUTING.

После создания правил фильтрации необходимо нажать на кнопку «**Применить конфигурацию**». Все изменения вступят в силу. Будьте особенно внимательны, не закрывайте доступ к порту 10000, на этом порту работает сервер `Webmin`.

Глава 13

Вопросы безопасности системы

Безопасность любой многопользовательской системы, в том числе и **ASPLinux**, включает два аспекта — ее сохранность на локальной машине (локальная безопасность) и защита от внешних воздействий (сетевая безопасность). Они связаны между собой, однако в настоящем руководстве будет рассмотрен только первый, локальный, аспект безопасности. Для изучения вопросов сетевой безопасности следует обратиться к дополнительным источникам информации (краткий обзор которых дан в заключении).

Безопасность локальной машины — это, в первую очередь, сохранность ее файловой системы. Она основывается на соблюдении некоторых несложных правил.

Первое из них — правильное завершение работы. В отличие от MS DOS (и, хотя и в меньшей степени, Windows 9x/ME), Linux-машину нельзя выключить с помощью выключателя электропитания, или перезагрузить с помощью «**Reset**». Вследствие эффективности кэширования дисковых операций, это почти гарантирует потерю данных, даже, казалось бы, сохраненных. Более того, «холодное» выключение с большей или меньшей вероятностью может повлечь за собой фатальное разрушение файловой системы.

Правда, ныне в Linux имеются довольно эффективные средства самовосстановления файловой системы при сбоях, в том числе и применение журналируемых файловых систем ext3 и Reiserfs. Однако риск все равно велик и, главное, неоправдан, поскольку избежать его несложно.

Именно, перед окончанием работы следует закрыть все работающие приложения, сохранив измененные файлы их штатными средствами (что, однако, пока не гарантирует, что они действительно будут записаны на диск), и выйти из системы X Window System, если она была запущена.

Далее возможно несколько вариантов завершения работы. При авторизации в качестве обычного пользователя, проще всего нажать комбинацию клавиш `Ctrl`+`Alt`+`Del` (в консоли). Компьютер пойдет на перезагрузку, и в момент

появления приглашения начального загрузчика (то есть выбора операционной системы) его можно безболезненно выключить. Если в качестве загрузчика используется ASPLoader, это делается через его меню *«File»*- *«Turn power off»*, что автоматически приводит к отключению питания (на материнских платах ATX).

Администратор системы для останова ее работы может воспользоваться командами halt или shutdown. Первая из них предназначена для немедленного завершения работы. По умолчанию она выполняет синхронизацию всех буферов, завершает работающие приложения (если таковые имеются) и делает запись в файле /var/log/wtmp. Выполненная с опцией -i, она предварительно завершает работу всех сетевых интерфейсов, с опцией -p — отключает питание компьютера после останова системы. Опция -n запрещает выполнение синхронизации.

Команда shutdown служит для корректного завершения работы через промежуток времени, указанный в качестве ее аргумента. Данная в форме, например,

```
shutdown +5
```

она остановит работу системы через пять минут. В форме

```
shutdown +0
```

остановит работу системы немедленно (эквивалентом последнего варианта является форма shutdown now). Время до останова системы может быть задано и в абсолютно формате чч:мм. В этом случае останов системы произойдет в указанный момент времени, например, по команде

```
shutdown 11:30
```

останов системы произойдет в 11 часов 30 минут.

Команда shutdown с опцией -r вызывает перезагрузку системы вместо ее останова. Именно она вызывается нажатием стандартной комбинации клавиш Ctrl+Alt+Del, что как уже говорилось, по умолчанию может выполнить любой пользователь (или просто человек, случайно получивший доступ к консоли). Такое поведение этой комбинации клавиш описано в файле /etc/inittab строкой вида

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -r now
```

Поэтому для предотвращения случайной перезагрузки пользователем достаточно удалить или закомментировать эту строку, после чего (во избежание перезапуска системы) выполнить команду init q.

Следует подчеркнуть, что процесс останова или перезагрузки Linux-машины занимает почти то же время, что и ее загрузка, поскольку все действия по монтированию файловой системы производятся при этом в обратном порядке.

Необходимость принудительного завершения сеанса может возникнуть при зависании программ. Обычно зависшая программа не препятствует ни вводу с клавиатуры (или переключению мышью), ни выводу на экран для других приложений. И при работе в системе X Window System достаточно открыть окно терминала (или строку минитерминала) и дать команду `xkill`, после чего щелкнуть мышью на окне зависшей программы. Окно это исчезнет, и система будет функционировать, как ни в чем не бывало. Хотя при этом зависшая программа может по прежнему «работать» в системе, загружая ресурсы компьютера.

Другой способ уничтожения зависшей программы (как в консоли, так и в X Window System) — использование команды `kill`. Для этого следует перейти в другую виртуальную консоль, с помощью команды `ps` установить `PID` зависшего процесса (или командой `jobs` — номер задания командной оболочки), а затем снять его командой

```
kill <PID>
```

или

```
kill %#
```

где # — номер задания оболочки. С помощью команды

```
kill <PID1> <PID2> ... <PID99>
```

можно одновременно снять любое количество процессов.

Крайне редко, но все же иногда возникает ситуация, когда программа `kill` не в состоянии уничтожить вышедший из-под контроля процесс (например, такой, в поле статуса которого стоит символ `D`). В этом случае можно попробовать вернуть управление системой с помощью `SysRq` (клавиша, совмещенная с `PrintScreen`), упоминавшейся в главе о компиляции ядра. Если ядро собрано с этой опцией, становятся доступными следующие клавишные комбинации:

- `Alt`+`SysRq`+`S` — немедленная синхронизация файловых систем, не портящая их при выключении питания;
- `Alt`+`SysRq`+`K` — завершение всех процессов, запущенных с текущей виртуальной консоли;

- Alt+SysRq+U — перемонтирование всех файловых систем в режим read only (только для чтения), что может помешать их порче;
- Alt+SysRq+B — немедленная перезагрузка системы, правда, без синхронизации и размонтирования файловых систем;
- Alt+SysRq+E — отправка сигнала SIGTERM (завершение с сохранением данных) всем процессам, кроме процесса init;
- Alt+SysRq+I — отправка сигнала SIGKILL (немедленное завершение) всем процессам, кроме процесса init;
- Alt+SysRq+L — отправка сигнала SIGKILL всем процессам, включая init, после чего система, естественно, становится нефункциональной.

Несмотря на широкий спектр средств управления процессами (в том числе и, казалось бы, зависшими), аварийное завершение работы исключить все же нельзя, хотя бы из-за сбоев питания. В этом случае в смонтированных файловых системах практически неизбежно возникают более или менее тяжелые ошибки, вызванные нарушением синхронизации буферов памяти, осуществляющих кэширование дисковых операций.

Для исправления таких ошибок предназначена программа fsck (от file system checker). Подобно scandisk в Windows, она автоматически запускается при старте компьютера после аварийного завершения (если, конечно, повреждения файловой системы не препятствуют загрузке вообще).

Программа fsck при старте отыскивает поврежденные участки файловой системы (обычно выражающиеся в нарушении связи между inode файла и его именем) и по возможности исправляет их автоматически. Если это оказывается ей не под силу, поврежденные фрагменты собираются в каталог /lost+found, где они могут быть просмотрены и, при удаче, частично восстановлены.

Программа fsck может быть запущена и вручную, после загрузки системы, что дает доступ к ее опциям. В качестве аргумента могут быть указаны имя устройства (/dev/hda#) или точка монтирования (/home). Основные опции команды следующие:

- -t type — явное указание типа проверяемой файловой системы (например, ext2 для файловой системы Linux);
- -A — проверка всех файловых систем, отмеченных в файле /etc/fstab;
- -R — в сочетании с предыдущей пропускает проверку корневой файловой системы;
- -a — производит автоматическое восстановление файловой системы (по умолчанию fsck работает интерактивно, с подтверждением предлагаемых действий);

- -V — выводит информацию о совершаемых действиях;
- -C — выводит шкалу хода проверки.

Как правило, безопаснее выполнять проверку файловых систем, не смонтированных в текущий момент. Для этого все они (кроме корневой, /), могут быть размонтированы командой umount с указанием имени устройства или точки монтирования в качестве аргумента. Однако с корневой системой так поступить не удастся, и потому, если требуется ее проверка, следует прибегнуть к альтернативному способу загрузки — со спасательной (resque) дискеты или дистрибутивного CD.

Спасательная дискета, как правило, изготовляется на стадии установки системы, хотя ее можно сделать и позднее. Однако еще проще для целей аварийного восстановления воспользоваться первым дистрибутивным CD, благо **ASPLinux**, в отличие от ряда других дистрибутивов, такую возможность предоставляет. Для этого требуется:

- войти в BIOS Setup и на основе руководства к материнской плате установить привод CD-ROM как первое загрузочное устройство;
- вставить первый CD из дистрибутива в соответствующий привод;
- перезагрузить компьютер, после чего запустится инсталляционная программа **ASPLinux**;
- дождавшись предложения выбрать язык установки, с помощью комбинации клавиш Alt+F2 переключиться на вторую виртуальную консоль.

После этого появится приглашение командной строки оболочки bash, но с несколько урезанными возможностями (в частности, отсутствием поддержки контроля заданий). Впрочем, для действий в аварийной ситуации имеющихся возможностей вполне хватает.

При таком способе загрузки, в отличие от обычного, корневая файловая система находится на виртуальном (RAM) диске, а все реальные, находящиеся на винчестере (винчестерах), файловые системы остаются несмонтированными. И на любой из них можно запустить программу проверки тем же образом, как это было описано выше. Кроме того, при загрузке с CD не запрашивается пароль администратора — система находится в т.н. однопользовательском режиме, о котором будет сказано чуть ниже.

Есть еще одна опасность — утерянный пароль пользователя или администратора. Это может случиться не только вследствие забывчивости, но и, например, при физическом повреждении файла /etc/passwd или /etc/shadow.

Первый случай (утрата пользовательского пароля) сложностей не доставляет: достаточно зайти в систему как администратор и сменить пользовательский па-

роль командой passwd с указанием имени пользователя, введя новый пароль и повторив его.

Несколько сложнее, если потерян пароль суперпользователя. В этом случае следует загрузиться со спасательной дискеты или дистрибутивного CD, как уже было рассказано выше, при необходимости смонтировать дисковый раздел, на котором расположен каталог /etc, и открыть в любом текстовом редакторе файл /etc/passwd. Первой строкой в нем будет учетная запись администратора:

```
root:xyz:0:0::/root:/bin/bash
```

где xyz между двумя первыми двоеточиями — некий набор символов, соответствующий зашифрованному паролю (с самим паролем он не имеет ничего общего, в том числе и количество символов не совпадает). Эта последовательность символов просто стирается, и после перезапуска системы уже нормальным образом авторизация в качестве суперпользователя введения пароля уже не требует: его следует заново определить командой passwd.

Однако такой способ сработает только в том случае, если не используются т. н. «теневые» (shadow) пароли (а в дистрибутиве **ASPLinux** по умолчанию используются именно они). Это сделано для повышения безопасности для машины, подключенной к любой (локальной или Глобальной) сети. Потому что файл /etc/passwd по умолчанию доступен для чтения любым пользователем, хотя право изменять его — только за администратором системы. Конечно, пароли в этом файле зашифрованные, но могут быть, если и не дешифрованы, то подобраны. Для предотвращения этого и придумана система «теневых» паролей: в этом случае в соответствующем поле файла passwd хранится только заменитель пароля, тогда как сам он расположен в файле /etc/shadow, который по умолчанию не доступен для чтения никому, кроме администратора. И где утерянный пароль и должен быть уничтожен с целью его последующей замены на новый.

Однако не обязательно действия по изменению пароля суперпользователя должны начинаться с загрузки с дискеты (или с CD) — на этот и аналогичные случаи предусмотрен так называемый однопользовательский режим. Чтобы прибегнуть к нему, при загрузке нужно в ответ на приглашение lilo выбрать загружаемое ядро (например, linux-2.4.x) и запустить его с параметром init=/bin/bash:

```
linux-2.4.x init=/bin/bash
```

В этом случае никакой пароль не запрашивается, и права администратора приобретаются автоматически. После чего можно производить любые действия по изменению настроек системы.

Глава 14

Заключение

В этом руководстве невозможно осветить все вопросы администрирования системы. Для получения дополнительной информации следует обратиться к специализированным книгам по системному администрированию Linux или UNIX. Вследствие совместимости **ASPLinux** с его прототипом — дистрибутивом RedHat, выбор какого либо из изданных в последнее время переводных руководств по администрированию последнего будет предпочтителен.

Кроме этого, ряд частных вопросов администрирования затрагивается во многих других книгах. Так, нетривиальные и полезные аспекты конфигурирования X Window System описаны в книге С.В. Зубкова «Linux. Русские версии» (М.: ДМК Пресс, 2000, с. 352).

Ряд уникальных, основанных на собственном опыте, сведений об администрировании Linux содержится в книге: В. Водолазкий, А. Колядов. «Путь к Linux». Изд. 2-е, пер. и доп. М.: Нолидж, 2001, 560 с. В частности, в ней большое внимание уделено проблемам сетевой безопасности.

Подробное описание таких ключевых понятий Linux, как устройство файловой системы, управление процессами и работа в командных оболочках, имеется в цикле статей Виктора Хименко:

Файлы, файлы, файлы. Мир ПК, 2000, № 2, с.64-68; № 3, с. 50-56,

Процессы, задачи, потоки и нити. Мир ПК, 2000, № 5, с. 42-47; № 6, с. 54-57,

Кто командует парадом. Мир ПК, 2001, № 1, с. 154-160; № 2, с. 151-156.

Большое количество информации по администрированию Linux можно подчерпнуть в журнале «Системный Администратор» `http://www.samag.ru`.

Множество ссылок на сайты с материалами по администрированию Linux можно найти на сайте `http://www.opennet.ru`.

Глава 15

Авторы документации

2007-07-31 13:33:48

Официальная документация **ASPLinux** написана в формате LaTeX. PostScript файл создан с помощью собственных файлов стилей.

Следующие люди приняли в той или иной мере участие в создании книги «*Быстрый старт*»:

Алексей Федорчук — первый автор книги;

Павел Гашев — создание стилей, множественные исправления и дополнения материала;

Коллектив авторов благодарит всех пользователей, внесших свой непосредственный вклад в исправление и дополнение документации.[1]

[1]Исправления, дополнения и пожелания принимаются по адресу http://bugzilla.asplinux.ru/ в разделе ASPLinux Documentation.

ДЛЯ ЗАМЕТОК

Заказ Г-3476 Отпечатано с оригиналов заказчика

Казанский производственный комбинат программных средств

1 2 3 4 5 6 7 8 9 0



ASPLINUX

ASPLinux
Россия:
127051, г.Москва, ул. Нижняя Сыромятническая, дом 5, корпус 7, строение 2, офис 723.
Телефон: +7 (495) 995-40-37 (общие вопросы и сотрудничество)

Украина:
83003, Донецк, Проспект Ильича, д. 89; Тел. +38 (062) 385-87-23

Общие вопросы: info@asplinux.ru
Служба технической поддержки: support@asplinux.ru
http://www.asplinux.ru